Naval Research Laboratory

Monterey, CA 93943-5502

NRL/MR/7543--96-7227

Software User's Manual for the Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS)

TRACY HAACK

Atmospheric Dynamics & Prediction Branch
Marine Meteorology Division

November 1996

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

Agency Use Only *(Leave Blank)*.	2. Report Date. November 1996	3. Report Type and Dates Covered. Final

4. Title and Subtitle.
Software User's Manual for the Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS)

5. Funding Numbers.
PE O & MN
PN NA
AN DN 153-174

6. Author(s).
Tracy Haack

7. Performing Organization Name(s) and Address(es).
Naval Research Laboratory
Marine Meteorology Division
Monterey, CA 93943-5502

8. Performing Organization Reporting Number.
NRL/MR/7534- - 96- - 7227

8. Sponsoring/Monitoring Agency Name(s) and Address(es).
Naval Meteorology & Oceanography Command (CNMOC)

10. Sponsoring/Monitoring Agency Report Number.

11. Supplementary Notes.

12a. Distribution /Availability Statement.
Approved for public release; distribution unlimited.

12b. Distribution Code.

13. Abstract *(Maximum 200 words).*

The Navy has developed a nonhydrostatic fully coupled ocean/atmosphere model, The Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS), for predicting and simulating meteorological processes on meso to micro scales of motion. This document covers only the atmospheric portion and is based upon version 2.0 of the COAMPS code. The atmospheric model may be initialized with idealized fields or real data adjusted by observations. Detailed descriptions and diagrams of the COAMPS directory structure and coding framework is given to familiarize the user with the system. In addition, COAMPS numerous input parameters are defined and categorized to aid the user in manipulating the code to suit a variety of possible case studies. Finally, the user is taken through a step-by-step procedure for implementing and working with the system and for setting up and submitting a COAMPS model run.

14. Subject Terms. Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS)	15. Number of Pages. 88
	16. Price Code.

Security Classification of Report.	18. Security Classification of This Page.	19. Security Classification of Abstract.	20. Limitation of abstract.
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	Same as report

Table of Contents

	Page
1. Scope	1
2. Reference Documents	2
2.1. COAMPS Model Descriptions	2
2.2. COAMPS Simulations	2
2.3. Dynamics	3
2.4. Physics	3
2.5. Numerics	4
2.6. COAMPS Graphics	5
2.7. Analysis	5
3. System Overview	6
3.1. COAMPS General Features	6
3.1.1. Driver Programs and Namelists	6
3.1.2. Model Domain Structure	8
3.1.3. Initial Conditions and COAMPS Databases	12
3.1.4. First Guess Fields and Data Assimilation	14
3.2. COAMPS Directory Structures	16
3.3. COAMPS Code	19
3.3.1. Analysis	19
3.3.1.1. Idealized Data	23
3.3.1.2. Real Data	23
3.3.2. Forecast	25
3.3.2.1. Reading Initial Conditions	27
3.3.2.2. Model Integration: Forecast Loop and Atmospheric Model	30
3.3.2.3. Writing Output	35
4. Execution Procedure	39
4.1. Preparation Steps	39
4.2. COAMPS Input File: *Gridnl, Coamnl* and *Mvoinl* Namelists	41
4.2.1. Model Setup	41
4.2.1.1. Grid	41
4.2.1.2. Integration	42
4.2.1.3. Boundary Conditions	42
4.2.2. Case Study Setup	43
4.2.2.1. Analysis Fields	43
4.2.2.2. Model Physics	44
4.2.2.3. Numerical Damping	44
4.2.2.4. Additional Model Features	44
4.2.3. Data Manipulation	44
4.3. Submitting a COAMPS Job	45

List of Appendices

	Page
Appendix A: Sample makefile	A-1
Appendix B: Sample COAMPS user input file *nlist.input*	B-1
Appendix C: Input parameters: *gridnl* namelist	C-1
Appendix D: Input parameters: *coamnl* namelist	D-1
Appendix E: Three sample map projections, input parameters and model domains	E-1

List of Figures

		Page
Figure 1:	Flow chart of the COAMPS driver programs.	7
Figure 2:	Schematic representation of three horizontally nested domains.	9
Figure 3a:	Schematic representation of the vertical grid staggering.	10
Figure 3b:	Schematic representation of the horizontal grid staggering.	11
Figure 4:	Schematic representation of the COAMPS data assimilation update cycle	15
Figure 5:	COAMPS directory structure.	17
Figure 6:	Flow chart of input/output for the analysis routine.	20
Figure 7:	Flow chart of the analysis routine.	22
Figure 8:	Flow chart of input/output for the forecast routine.	26
Figure 9:	Flow chart of the forecast routine.	28
Figure 10:	Flow chart of the atmospheric model subroutine.	30
Figure 11:	Flow chart of the subroutines called by *coamm.f* that write out model data.	36
Figure E-1:	Schematic representation of a mercator map projection.	E-1
Figure E-2:	Example of a COAMPS horizontal domain using a mercator map projection.	E-2
Figure E-3:	Schematic representation of a lambert conformal map projection.	E-3
Figure E-4:	Example of a COAMPS horizontal domain using a lambert conformal map projection.	E-4
Figure E-5:	Schematic representation of a polar stereographic map projection.	E-5
Figure E-6:	Example of a COAMPS horizontal domain using a polar stereographic map projection.	E-6

List of Tables

		Page
Table 1a:	COAMPS 2D surface arrays.	12
Table 1b:	COAMPS 3D basic state arrays.	13
Table 1c:	COAMPS 3D prognostic arrays.	13
Table 2:	COAMPS array naming convention.	14
Table 3:	Standard machine settings for COAMPS compilation.	18
Table 4:	COAMPS output file naming convention.	37
Table 5:	Sigma level data and surface fields written out by COAMPS.	38
Table 6:	Options for the 2nd run script input argument.	45
Table D-1:	Namelist input options for the TKE mixing length parameter **iamxgl**.	D-1
Table E-1:	Namelist input parameters and values for the mercator map projection shown in Figure E-2.	E-1
Table E-2:	Namelist input parameters and values for the lambert conformal map projection shown in Figure E-4.	E-3
Table E-3:	Namelist input parameters and values for the polar stereographic map projection shown in Figure E-6.	E-5

Acknowledgments

Many helpful suggestions were given by reviewers Dr. Richard Hodur, Dr. William Thompson, Ms. Liana Zambresky, Dr. Teddy Holt and Ms. Patricia Phoebus of NRL Monterey. The assistance of Ms. Sue Chen in reorganizing the document is also gratefully acknowledged. This work was sponsored by the Oceanographer of the Navy, and the Program Office at Space and Naval Warfare Systems Command (PMW-185), under program element 0603207N.

1. Scope

Numerical models have been an effective tool in the prediction of many geophysical systems. Processes within the earth's two primary physical systems, the ocean and atmosphere, directly impact Naval operations on the mesoscale and the local PBL scale. Consequently, accurate and efficient prediction on these scales is a necessity. Scientists now consider the ocean and atmosphere as separate but fully coupled, two-way interactive fluids. Predicting the behavior of either fluid depends upon the spatial and temporal forcing applied by the other. Thus, a single numerical prediction system combining an oceanic and an atmospheric model provides more realistic representation of these two geophysical systems. Additionally in the past decade, increased computer power and technological advancements have improved computational efficiency allowing larger models, with higher resolution, multi-nested grids and complicated physics, to be developed and run for real-time forecasting purposes. When used in a research mode, the models also provide valuable insight toward understanding complex mesoscale interactions. To fully utilize modern computer resources and to meet the growing need for high resolution, coupled oceanic/atmospheric forecasts, a new model has been developed by the Naval Research Laboratory: The Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS).

At present, COAMPS consists of two FORTRAN programs: an analysis program that blends observations with first guess fields to provide initial conditions, and a forecast program containing a nonhydrostatic, quasi-compressible atmospheric model with complete physics schemes for predicting meso and micro scales of motion (i.e. time scales ranging from days to minutes and spatial scales ranging from thousands of kilometers to a few meters). This system can be run using idealized or real data initial conditions with up to seven horizontally nested domains. Because the ocean forecast model of COAMPS currently is under development, it will be implemented at a later date. For this reason, this report documents only the atmospheric model. Moreover, the focus here is to provide a brief introduction to COAMPS, along with a quick reference guide for setting up and performing a COAMPS model run. It is not intended as a complete description of the model physics or dynamics. Additional reading material on COAMPS and related references are given in Section 2. An overview of the COAMPS directory structures and the code is given in Section 3. In Section 4. the procedure for executing a model run is described for either idealized or real data assimilation process studies. This section is beneficial in assisting the new user and also as a reference guide for identifying, defining and modifying relevant model parameters. Appendices A-E provide sample files and tables that list and define COAMPS input parameters.

2. Reference Documents

2.1. COAMPS Model Descriptions

Haack, T.H., 1993: User's guide for the coupled ocean/atmosphere mesoscale prediction system (COAMPS). *NRL Memorandum Report 7214,* November, 1993.

Hodur, R.M., 1993: Development and testing of the coupled ocean/atmosphere mesoscale prediction system (COAMPS). *NRL Memorandum Report 7213,* November, 1993.

Hodur, R.M.: The U.S. Navy's coupled ocean/atmosphere mesoscale prediction system (COAMPS). *Submitted to Mon. Wea. Rev.*

Hodur, R.M. and J. Doyle, 1995: The coupled ocean/atmosphere mesoscale prediction system (COAMPS). Chapter in "Coastal Ocean Prediction", CRC Press, Boca Raton, Florida, 96 pp.

Xu, L., 1995: Incorporation of orography into the coupled ocean/atmosphere mesoscale prediction system (COAMPS): formulation and validation tests. Chapter 3 in "The study of mesoscale land-air-sea interaction processes using a nonhydrostatic model", Ph.D. dissertation, North Carolina State University, 336pp.

2.2. COAMPS Simulations

Doyle, J., 1995: Coupled ocean wave/atmosphere mesoscale model simulations of cyclogenesis. *Tellus,* **47A,5:1**, 766-778.

Hirschberg, P.A. and J. Doyle, 1995: An examination of pressure tendency mechanisms in an idealized simulation of extratropical cyclogenesis. *Tellus,* **47A,5:1**, 747-758.

Hodur, R.M.: Development and testing of the coupled ocean/atmosphere mesoscale prediction system (COAMPS). *NRL Memorandum Report 7533,* November, 1993.

Thompson, W.T., T. Haack and J.D. Doyle, 1996: An investigation of the southerly surge. *1996 NRL Review,* 151-153.

Thompson, W.T., T. Haack, J.D. Doyle and S.D. Burk, 1996: A nonhydrostatic mesoscale simulation of the 10-11 June 1994 trapped coastal wind reversal. *Submitted to Mon. Wea. Rev.*

Xu, L., S. Raman, R.V. Madala and R. Hodur, 1996: A non-hydrostatic modeling study of surface moisture effects on mesoscale convection induced by sea breeze circulation. *Meteorol. Atmos. Phys.*, **58**, 103-122.

2.3. Dynamics

Klemp, J.B. and R.B. Wilhelmson, 1978: The simulation of three-dimensional convective storm dynamics. *J. Atmos. Sci.*, **35**, 1070-1096.

2.4. Physics

Businger, J.A., J.C. Wyngaard, Y. Izumi, and E.F. Bradley, 1971: Flux-profile relationships in the atmospheric surface layer. *J. Atmos. Sci.*, **28**, 181-189.

Deardorff, J.W., 1972: Numerical investigation of neutral and unstable planetary boundary layers. *J. Atmos. Sci.*, **29**, 91-115.

Deardorff, J.W., 1978: Efficient prediction of ground surface temperature and moisture with inclusion of a layer of vegetation. *J. Geophys. Res.*, **83**, 1889-1903.

Deardorff, J.W., 1980: Stratocumulus-capped mixed layers derived from a three-dimensional model. *Bound. Layer Meteor.*, **18**, 495-527.

Hogan, T.F. and T.E. Rosmond, 1991: The description of navy operational global atmospheric prediction system's spectral forecast model. *Mon. Wea. Rev.*, **119**, 1786-1815.

Harshvardhan, R.D., D. Randall and T. Corsetti, 1978: A fast radiation parameterization for atmospheric circulation models. *J. Geophys. Res.*, **92**, 1009-1016.

Louis, J.F., 1979: A parametric model of vertical eddy fluxes in the atmosphere. *Bound. Layer Meteor.*, **17**, 187-202.

Louis, J.F., M. Tiedtke and J.F. Geleyn, 1982: A short history of the operational PBL-parameterization at ECMWF, *Workshop on Planetary Boundary Parameterization*, ECMWF, Reading, 59-79.

Mellor, G.L. and T. Yamada, 1974: A hierarchy of turbulence closure models for planetary boundary layers. *J. Atmos. Sci.*, **31**, 1791-1806.

Mellor, G.L. and T. Yamada, 1982: Development of a turbulence closure for geophysical fluid problems. *Rev. Geophys and Space Phys.*, **20**, 851-875.

Rutledge S.A. and P.V. Hobbs, 1983: The mesoscale and microscale structure of organization of clouds and precipitation in midlatitude cyclones. VIII: A model for the "seeder-feeder" in warm-frontal rainbands. *J. Atmos. Sci.*, **40**, 1185-1206.

Smagorinsky, J., 1963: General circulation experiments with the primitive equations: 1. The basic experiment. *Mon. Wea. Rev.*, **91**, 99-164.

Therry, G. and T. LaCarrere, 1983: Improving the eddy kinetic energy model for planetary boundary layer description. *Bound. Layer Meteor.*, **25**, 63-88.

2.5. Numerics

Arakawa, A. and V.R. Lamb, 1974: *Computational design of the UCLA general circulation model.* Methods in Computational Physics, Vol. 17, Academic Press, 173-265.

Chorin, A.J., 1967: A numerical method for solving incompressible viscous flow problems. *J. Comput. Phys.*, **2**, 12-16.

Cressman, G., 1959: An operational objective analysis system. *Mon. Wea. Rev.*, **87**, 1367-374.

Dudhia, J., 1993: A nonhydrostatic version of the Penn State-NCAR mesoscale model: validation tests and simulation of an Atlantic cyclone and cold front. *Mon. Wea. Rev.*, **121**, 1493-1513.

Durran D.R. and J.B. Klemp, 1983: A compressible model for the simulation of moist mountain waves. *Mon. Wea. Rev.*, **111**, 2341-2361.

Droegemeier, K.K. and R.B. Wilhelmson, 1987: Numerical simulation of thunderstorm outflow dynamics. Part I: Outflow sensitivity experiments and turbulence dynamics. *J. Atmos. Sci.*, **44**, 1180-1210.

Haltiner, G.J. and R.T. Williams, 1980: *Numerical prediction and dynamic meteorology.* John Wiley & Sons, Inc. 477 pp.

Hodur, R.M., 1987: Evaluation of a regional model with an update cycle. *Mon. Wea. Rev.*, **115**, 2707-2718.

Ikawa, M., 1988: Comparison of some schemes for nonhydrostatic models with orography. *J. Meteor. Soc. Japan*, **66**, 753-776.

Klemp, J.B. and D.K. Lilly, 1978: Numerical simulation of hydrostatic mountain waves. *J. Atmos. Sci.*, **35**, 78-107.

Klemp, J.B. and R.B. Wilhelmson, 1978: The simulation of three-dimensional convective storm dynamics. *J. Atmos. Sci.*, **35**, 1070-1096.

Miller, M.J. and A.J. Thorpe, 1981: Radiation conditions for the lateral boundaries of limited area numerical models. *Quart. J. R. Met. Soc.*, **107**, 615-628.

Orlanski, I., 1976: A simple boundary condition for unbounded hyperbolic flows. *J. Comput. Phys.*, **21**, 251-269.

Robert, A.J., 1966: The investigation of a low order spectral form of the primitive meteorological equations. *J. Meteor. Soc. Japan*, **44**, 237-245.

Shamarock, W.C. and J.B. Klemp, 1992: The stability of time-split numerical methods for the hydrostatic and the nonhydrostatic elastic equations. *Mon. Wea. Rev.*, **120**, 2109-2127.

2.6. COAMPS Graphics

Haack, T.H., 1992: Graphical display procedure for grid point model data. *NRL Technical Note 263*, June, 1992.

Lewit, H. (CSC), 1993: Description and use of the ocards process management system. *FNOC Model Dept. Tech. Note 4-93*, September, 1993.

2.7. Analysis

Baker, N.L, 1992: Quality control for the navy operational atmospheric database. *Wea. Forecasting*, **7**, 250-261.

Barker, E.H, 1992: Design of the navy's multivariate optimum interpolation analysis system. *Wea. Forecasting*, **7**, 527-528.

Goerss, J.S. and P.A. Phoebus, 1992: The Navy's operational atmospheric analysis. *Wea. Forecasting*, **7**, 232-249.

3. System Overview

To effectively utilize all the features of COAMPS, the user must become familiar with two aspects of the system: the COAMPS directory structure and coding framework, and the process for submitting a COAMPS job. While the latter subject is covered in Section 4, here a brief overview of the complete modeling system is provided. The initial COAMPS directory tree consists of several subdirectories that contain the COAMPS source code, databases, templates for setting up a COAMPS job, documentation, and utility programs. Discussion of this directory structure is followed by an outline of the COAMPS coding which is made up of an analysis and forecast driver program. The use of flow chart diagrams and subroutine descriptions provides an overview of the two programs. Both the COAMPS directory structure and coding framework are of critical importance in understanding the modeling system, working with the code and properly executing a job.

The following conventions are adopted throughout the document: all COAMPS variables (including inputs, array names and other model parameters) are in boldface type. Appendix C, D and E lists and defines each of these model input parameters and is to be used as a reference while reading this document. Filenames (including subroutines, executables, script files, and namelists) are in italics. Some filenames contain a '$' indicating variable substitution. Before describing the COAMPS directory structure and coding, a few important general features are introduced below.

3.1. COAMPS General Features

The COAMPS code is more effectively outlined by first summarizing a few details about the model. Introduced in the following paragraphs are the driver programs, namelists, model domain, data assimilation procedure and COAMPS databases.

3.1.1. Driver Programs and Namelists

Because COAMPS is a large system, comprising nearly 83,000 lines of FORTRAN code, information can best be conveyed by flow chart diagrams -- the first of which is shown in Figure 1 for the COAMPS analysis and forecast driver programs, *coamps_analysis.F* and *coamps_forecast.F* respectively. The driver program's function is to read in the user defined model grid dimensions that provide array allocation space and pointer locations for all of COAMPS spatially dependent variables. These variables consist of the number of grid nests (**nnest**), the number of horizontal grid points for each nest (**ma(nn), na(nn)**, where **nn** is the nest number), and the number of vertical levels (**kka**). Their values are specified by reading in the namelist *gridnl*. Then subroutine *mema.F* is called to compute the total length, **nwords**, of a single array (**a**) that allocates array space for all the analysis fields on all meshes. In addition to computing the length of array **a**, *mema.F* also assigns the pointer location within that array for each field on each nest. These pointer locations are denoted by the variable

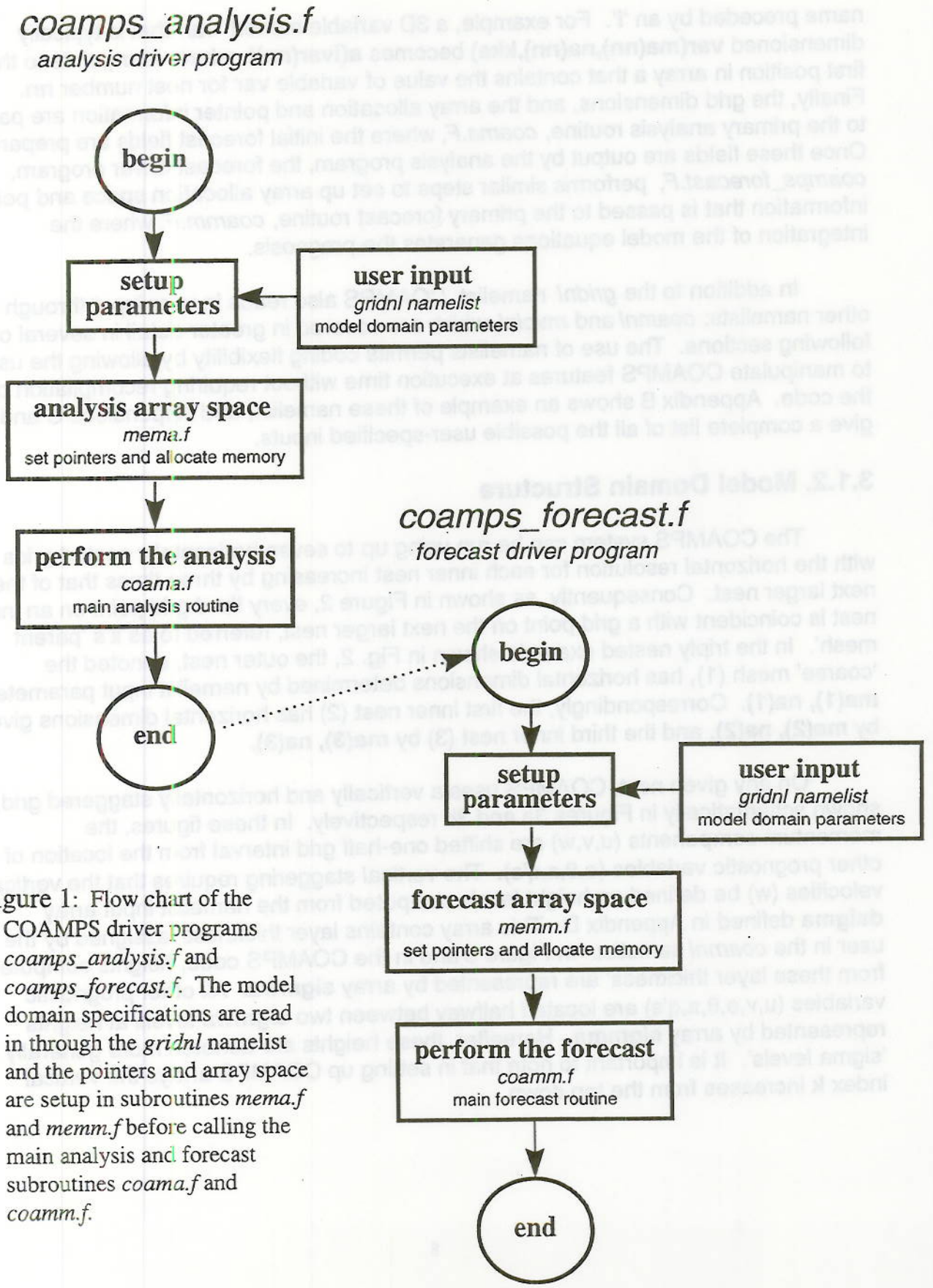


Figure 1: Flow chart of the COAMPS driver programs *coamps_analysis.f* and *coamps_forecast.f*. The model domain specifications are read in through the *gridnl* namelist and the pointers and array space are setup in subroutines *mema.f* and *memm.f* before calling the main analysis and forecast subroutines *coama.f* and *coamm.f*.

name preceded by an **'i'**. For example, a 3D variable named **'var'** that is typically dimensioned **var(ma(nn),na(nn),kka)** becomes **a(ivar(nn))**, where **ivar** points to the first position in array **a** that contains the value of variable **var** for nest number **nn**. Finally, the grid dimensions, and the array allocation and pointer information are passed to the primary analysis routine, *coama.F*, where the initial forecast fields are prepared. Once these fields are output by the analysis program, the forecast driver program, *coamps_forecast.F*, performs similar steps to set up array allocation space and pointer information that is passed to the primary forecast routine, *coamm.F*, where the integration of the model equations generates the prognosis.

In addition to the *gridnl* namelist, COAMPS also reads in user input through two other namelists: *coamnl* and *mvoinl*, which are covered in greater detail in several of the following sections. The use of namelists permits coding flexibility by allowing the user to manipulate COAMPS features at execution time without requiring recompilation of the code. Appendix B shows an example of these namelist, and Appendices C and D give a complete list of all the possible user-specified inputs.

3.1.2. Model Domain Structure

The COAMPS system can be run using up to seven horizontally nested grids with the horizontal resolution for each inner nest increasing by three times that of the next larger nest. Consequently, as shown in Figure 2, every third grid point on an inner nest is coincident with a grid point on the next larger nest, referred to as it's 'parent mesh'. In the triply nested example shown in Fig. 2, the outer nest, denoted the 'coarse' mesh (1), has horizontal dimensions determined by namelist input parameters **ma(1), na(1)**. Correspondingly, the first inner nest (2) has horizontal dimensions given by **ma(2), na(2)**, and the third inner nest (3) by **ma(3), na(3)**.

On any given nest, COAMPS uses a vertically and horizontally staggered grid, shown schematically in Figures 3a and 3b respectively. In these figures, the momentum components (u,v,w) are shifted one-half grid interval from the location of the other prognostic variables (e,θ,π,q's). The vertical staggering requires that the vertical velocities (w) be defined on height levels computed from the namelist input array **dsigma** defined in Appendix D. This array contains layer thickness' assigned by the user in the *coamnl* namelist. In Figure 3 and in the COAMPS code, heights computed from these layer thickness' are represented by array **sigmwa.** All other prognostic variables (u,v,e,θ,π,q's) are located halfway between two **sigmwa** levels at heights represented by array **sigmma**. Hereafter, these heights are denoted more generally as 'sigma levels'. It is important to note that in setting up COAMPS arrays the vertical index **k** increases from the top down.

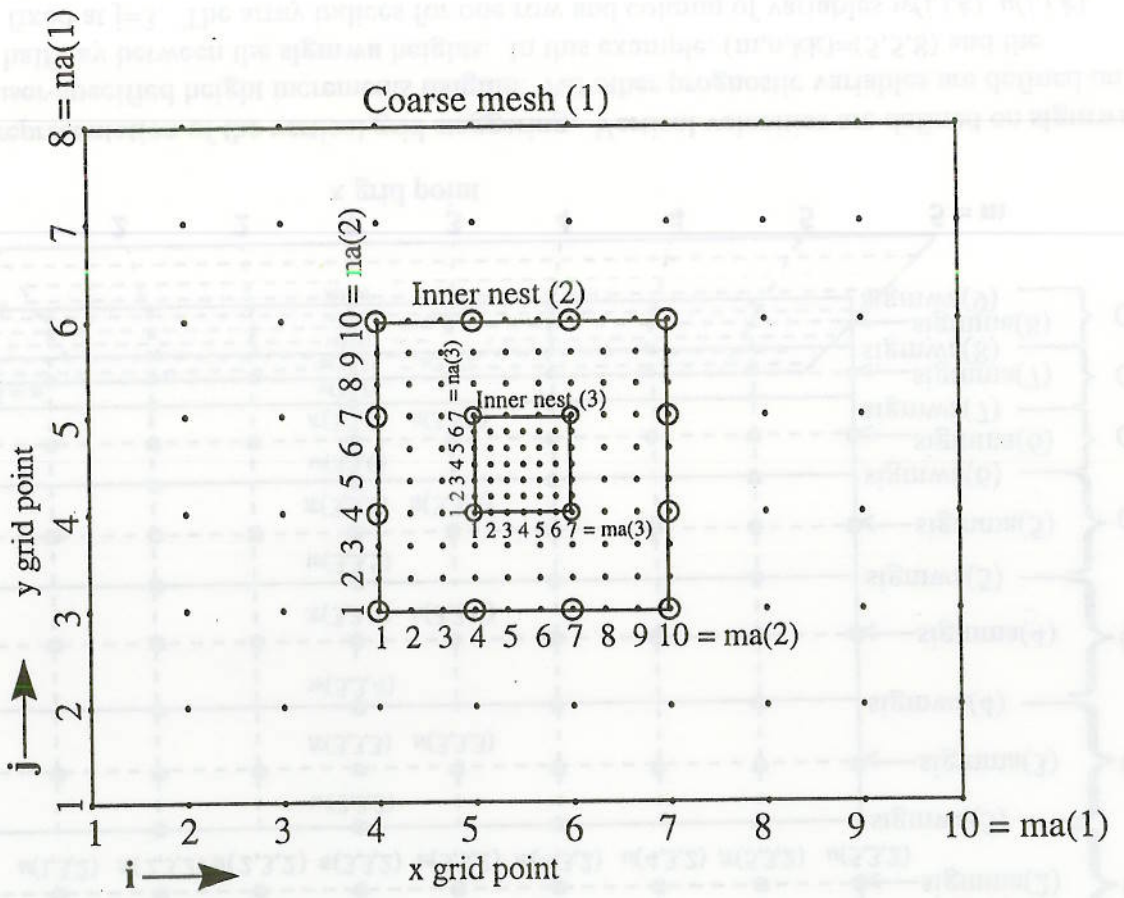


Figure 2: Schematic representation of three horizontally nested domains. The outer coarse mesh (1) contains horizontal grid points **ma(1)**=10 and **na(1)**=8. The first inner nest (2) contains horizontal grid points **ma(2)**=10 and **na(2)**=10. The second inner nest (3) contains horizontal grid points **ma(3)**=7 and **na(3)**=7. Note that every third inner nest grid point is collocated with a grid point on it's parent nest, denoted by the open circles. Thus, the number of inner nest grid points -1 must be divisible by 3 (e.g. (**ma(2)**-1)/3=3).

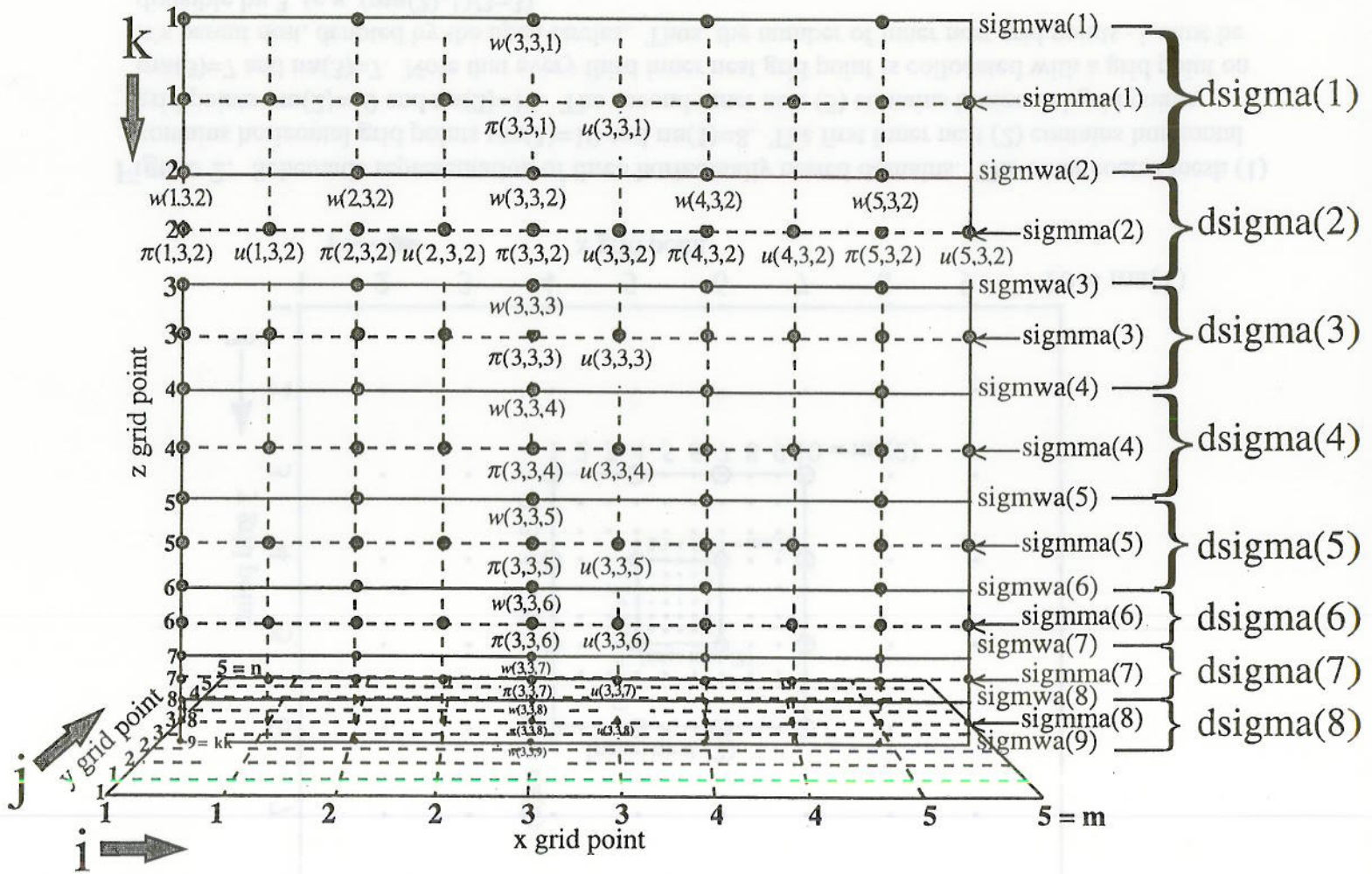


Figure 3a: Schematic representation of the vertical grid staggering. Vertical velocities are defined on **sigmwa** levels computed from user-specified height increments **dsigma**. All other prognostic variables are defined on **sigmma** levels located halfway between the **sigmwa** heights. In this example, (m,n,kk)=(5,5,8) and the vertical plane shown is fixed at j=3. The array indices for one row and column of variables *w(i,j,k), u(i,j,k), π(i,j,k)* are shown for reference, where *π*, the dimensionless pressure, is representative of the other mass variables.

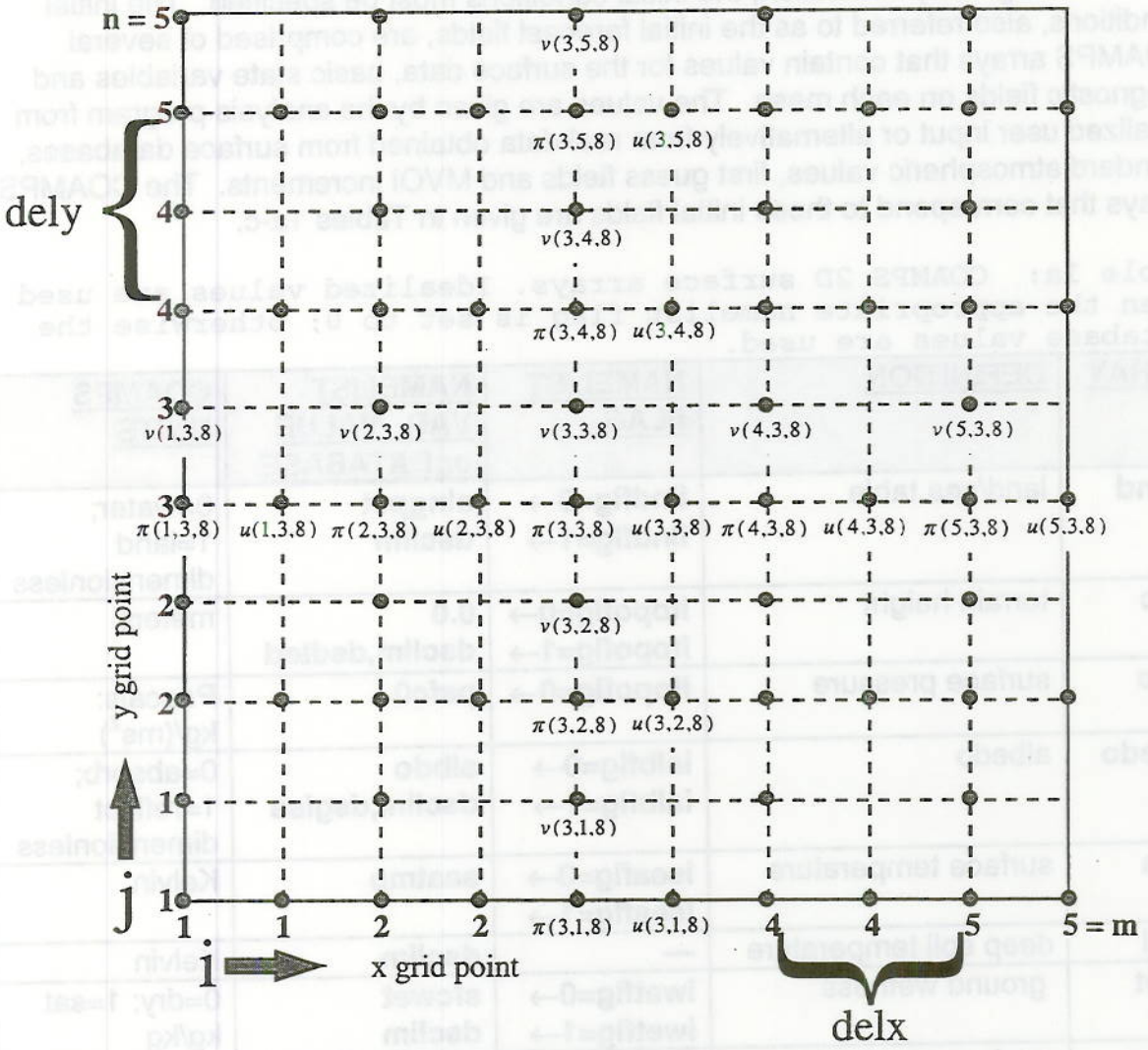


Figure 3b: Schematic representation of the horizontal grid staggering. The u- and v-momentum components are shifted one-half grid interval from the other prognostic variables. The **delx** and **dely** are the x and y grid intervals specified by the user for the coarse mesh. In this example, (m,n,kk)=(5,5,8) and the horizontal plane shown is fixed at k=8. The array indices for one row and column of variables *u(i,j,k), v(i,j,k),*π*(i,j,k)* areshown for reference, where π, the dimensionless pressure, is representative of the other mass variables.

3.1.3. Initial Conditions and COAMPS Databases

To begin any simulation, the initial conditions must be specified. The initial conditions, also referred to as the initial forecast fields, are comprised of several COAMPS arrays that contain values for the surface data, basic state variables and prognostic fields on each mesh. The values are given by the analysis program from idealized user input or alternatively from real data obtained from surface databases, standard atmospheric values, first guess fields and MVOI increments. The COAMPS arrays that correspond to these initial fields are given in Tables 1a-c.

Table 1a: COAMPS 2D surface arrays. Idealized values are used when the appropriate namelist flag is set to 0; otherwise the database values are used.

ARRAY	DEFINITION	NAMELIST FLAG	NAMELIST VAR, VALUE or DATABASE	COAMPS UNITS
xland	land/sea table	**ilndflg=0→** **ilndflg=1→**	**alngpct** **dsclim**	0=water; 1=land dimensionless
zsfc	terrain height	**itopoflg=0→** **itopoflg=1→**	**0.0** **dsclim,dsdted**	meters
psfc	surface pressure	**itopoflg=0→**	**psfc0**	Pascals: $kg/(ms^2)$
albedo	albedo	**ialbflg=0→** **ialbflg=1→**	**albdo** **dsclim,dsgiss**	0=absorb; 1=reflect dimensionless
tsea	surface temperature	**iseaflg=0→** **iseaflg=1→**	**seatmp**	Kelvin
tsoil	deep soil temperature	**---**	**dsclim**	Kelvin
gwet	ground wetness	**iwetflg=0→** **iwetflg=1→**	**sfcwet** **dsclim**	0=dry; 1=sat kg/kg
z0	surface roughness.	**iz0flg=0→** **iz0flg=1→**	**zrough** **dsclim,dsgiss**	meters

Table 1b: COAMPS 3D basic state arrays. Idealized values are used when namelist variable **icase**≠0; Standard atmospheric values are used when **icase**=0.

ARRAY	DEFINITION	NAMELIST FLAG	NAMELIST VAR or VALUE	COAMPS UNITS (NAMELIST VAR)
ugeoa	geostrophic u-wind component	**icase≠0→** **icase=0→**	**usnd & umean** **ugeoa**	m/s (**iuvsnd=1**) dir/spd (**iuvsnd=2**)
vgeoa	geostrophic v-wind component	**icase≠0→** **icase=0→**	**vsnd & vmean** **vgeoa**	m/s (**iuvsnd=1**) dir/spd (**iuvsnd=2**)
tbw/tbm	temperature	**icase≠0→**	**tsnd**	Kelvin (**itsnd=1**)

tbw/tbm	temperature	**icase=0→**	standard atmos	Celsius (**itsnd=2**) theta (**itsnd=3**)
qvbw & qvbm	specific humidity	**icase≠0→** **icase=0→**	**qsnd** computed from standard atmos	dewpt K (**iqsnd=1**) dewpt C (**iqsnd=2**) g/kg (**iqsnd=3**) rel hum (**iqsnd=4**)
prbw & prbm	pressure	**icase≠0→** **icase=0→**	**ipsnd** standard atmos	Pascal: $kg/(ms^2)$
exbw & exbm	dimensionless pressure	---	computed internally	Pascal: $kg/(ms^2)$
thbw & thbm	potential temperature	---	computed internally	Kelvin
thvbw & thvbm	virtual potential temperature	---	computed internally	Kelvin
rbw & rbm	density	---	computed internally	kg/m^3

Table 1c: COAMPS 3D prognostic arrays. Idealized values are used when namelist variable **icase**≠0; First guess fields (and MVOI increments when **iupd**=2) are used when **icase**=0.

ARRAY	DEFINITION	NAMELIST FLAG	NAMELIST VAR or VALUE	UNITS
u1	u momentum	**icase≠0→** **icase=0→** **iupd=2→**	**usnd** & **umean** first guess field & MVOI increments	m/s
v1	v momentum	**icase≠0→** **icase=0→** **iupd=2→**	**vsnd** & **vmean** first guess field & MVOI increments	m/s
w1	vertical velocity	**icase≠0→** **icase=0→**	**0.0** first guess field	m/s
th1	potential temperature (θ)	**icase≠0→** **icase=0→** **iupd=2→**	**thbm** first guess field & MVOI increments	Kelvin
e1	turbulent kinetic energy (TKE)	**icase≠0→** **icase=0→**	**1.0e-10** first guess field	m^2/s^2
qv1	mixing ratio: water vapor	**icase≠0→** **icase=0→** **iupd=2→**	**qvbm** first guess field & MVOI increments	kg/kg
qc1/qi1/ qr1/qs1	mixing ratios: clouds/ice/rain/snow	**icase≠0→** **icase=0→**	**0.0** first guess field	kg/kg
p1	dimensionless pressure perturbation (π)	**icase≠0→** **icase=0→** **iupd=2→**	**0.0** first guess field & MVOI increments	kg/ms^2

Because COAMPS uses the leap-frog integration scheme, three time levels for each of the 3D prognostic fields must be retained during each iteration. Thus, the array names are further identified by a time level number 1, 2, or 3:

Table 2: COAMPS array naming convention

NUMBER	TIME LEVEL
1	previous (t-Δt)
2	present (t)
3	predicted (t+Δt)

Note that the 3D arrays shown in Table 1c are for time level 1 representing values from the previous iteration. For purposes of discussion, the arrays are denoted here and throughout the remainder of the document by the conventional notation **vartl(ma(1),na(1),kka)**. In the code, the 3D prognostic arrays are denoted by **a(ivartl(nn))**, where '**a**' is the forecast program array, '**i**' is the pointer location for variable '**var**' at time level '**tl**' on nest number '**nn**'. As mentioned, the array dimension parameters (**ma(nn), na(nn),kka**) correspond to the number of grid points for each mesh that are specified by the user in the *gridnl* namelist. In general, all of the above arrays, along with other fields and variables in COAMPS, are in MKS units. With these preliminary details in mind, the COAMPS directory structure is described below.

3.1.4. First Guess Fields and Data Assimilation

Real data initial conditions require a set of first guess fields that represent the initial state of the atmosphere. These fields are given in one of two ways: from a NOGAPS analysis or forecast, defined as a 'NOGAPS cold start', or from a previous COAMPS forecast, defined as a 'data assimilation update cycle'.

The NOGAPS cold start provides global fields on pressure levels and at 1° horizontal resolution that are interpolated to the model's horizontal grid points. These fields are either initial conditions to the forecast program, or optionally, a first guess to the analysis program where a multi-variate optimum interpolation (MVOI) scheme blends in observational data.

The data assimilation update cycle, shown schematically in Figure 4, begins with a NOGAPS cold start and an MVOI analysis to produce COAMPS forecast fields valid one or two days prior to a particular study period. These forecast fields or 'history fields', on the model's sigma levels and at the model's horizontal grid points, are a first guess to the analysis program for the next COAMPS forecast. The assimilation cycle continues until history fields are produced for the simulation study period. Thus, the mesoscale character of the flow is generated and maintained by implementing a data assimilation update cycle.

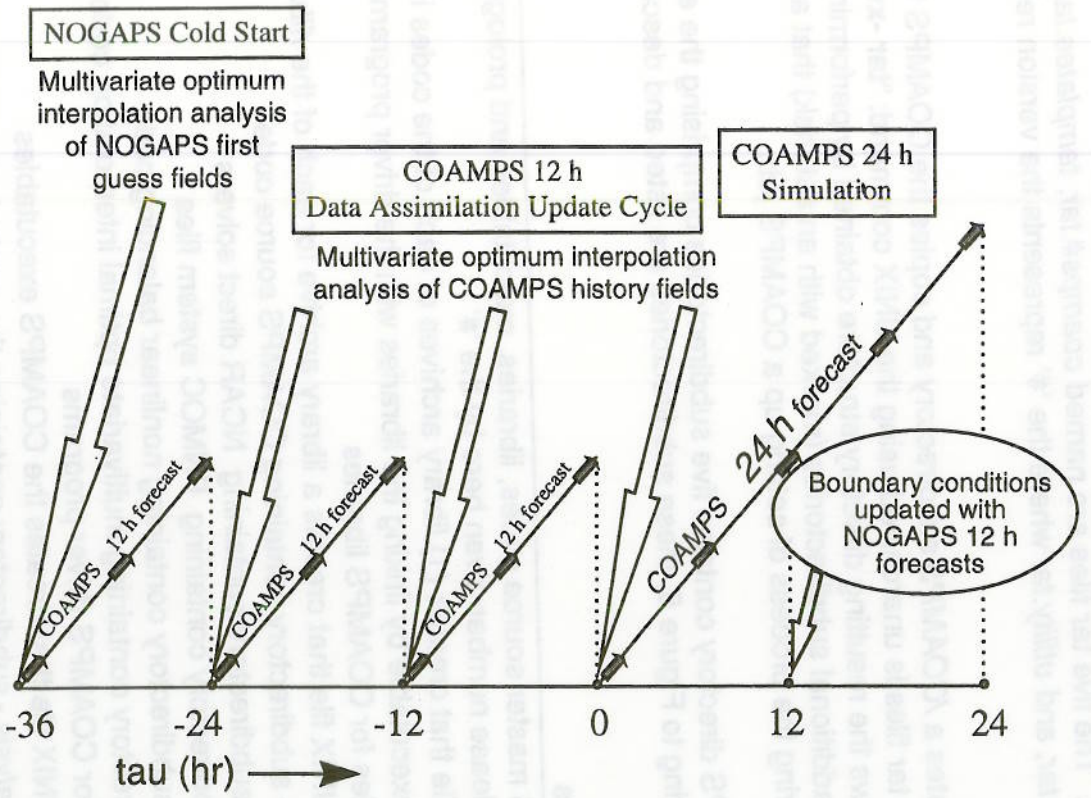


Figure 4: Schematic representation of the COAMPS data assimilation update cycle. The procedure begins with a 12-hour NOGAPS cold start. In this example, the cold start is initiated 36 hours before the initiation of the study period at tau=0 h. The cold start is followed by two 12-hour data assimilation updates using COAMPS forecast fields. A third data assimilation update initiates the 24-hour forecast for the simulation study period. Boundary conditions during the simulation are supplied by NOGAPS forecasts every 12 hours.

3.2. COAMPS Directory Structures

The COAMPS code can be run on several different UNIX platforms including the CRAY, SGI, DEC ALPHA, and HP machines. The process for implementing COAMPS involves first obtaining the five COAMPS tar files. Requests for COAMPS must be submitted to Dr. Richard Hodur, email: **hodur@nrlmry.navy.mil**. Also forward a copy of the request to the COAMPS system administrator Sue Chen, email: **chen@nrlmry.navy.mil**. The five tar files are named *coamps#.tar*, *templates.tar*, *database.tar*, *document.tar*, and *utility.tar* where the '#' represents the version release number.

The user first creates a */COAMPS* subdirectory and copies the COAMPS tar files into it. Then each of the tar files is unarchived using the UNIX command: "**tar -xvf** *tar filename*". Figure 5 shows the resulting directory structure obtained by performing this step, along with several additional subdirectories (marked with an asterisk) that are automatically created during the process of setting up a COAMPS job.

The initial COAMPS directory contains five subdirectories comprising the entire COAMPS system. Referring to Figure 5, these subdirectories are listed and described below:

COAMPS Subdirectories

- */coamps#* - COAMPS master source codes, libraries, executables and prologues for a particular version release number given here by the '#'
  - */Makefile* - UNIX file that creates (1) library archives of each of the codes in */libsrc*, and (2) an executable by linking the libraries with the driver programs in */src*
  - */libsrc* - source codes for COAMPS libraries
    - */Makefile* - UNIX file that creates a library archive for each of the */libsrc* codes
    - */coampslib* - subdirectory containing COAMPS source code
    - */fishpaklib* - subdirectory containing NCAR direct solvers
    - */fnoclib* - subdirectory containing FNMOC system files
    - */nl_beqlib* - subdirectory containing nonlinear balancing code
    - */oilib* - subdirectory containing multivariate optimal interpolation code
  - */src* - source code for COAMPS driver programs
    - */Makefile* - UNIX file that creates the COAMPS executables
    - */coamps_analysis* - subdirectory containing the analysis driver program
    - */coamps_forecast* - subdirectory containing the forecast driver program
    - */newdtg* - subdirectory containing the date-time group program
  - */prologues* - log files for describing source codes and tracking code modifications
  - **(The following additional subdirectories (*/lib, /bin*) are created by the *Makefile* in subdirectory */coamps#*)**
  - */lib* - location of library archives from each of the codes in */libsrc*
  - */bin* - location of executables for COAMPS master code

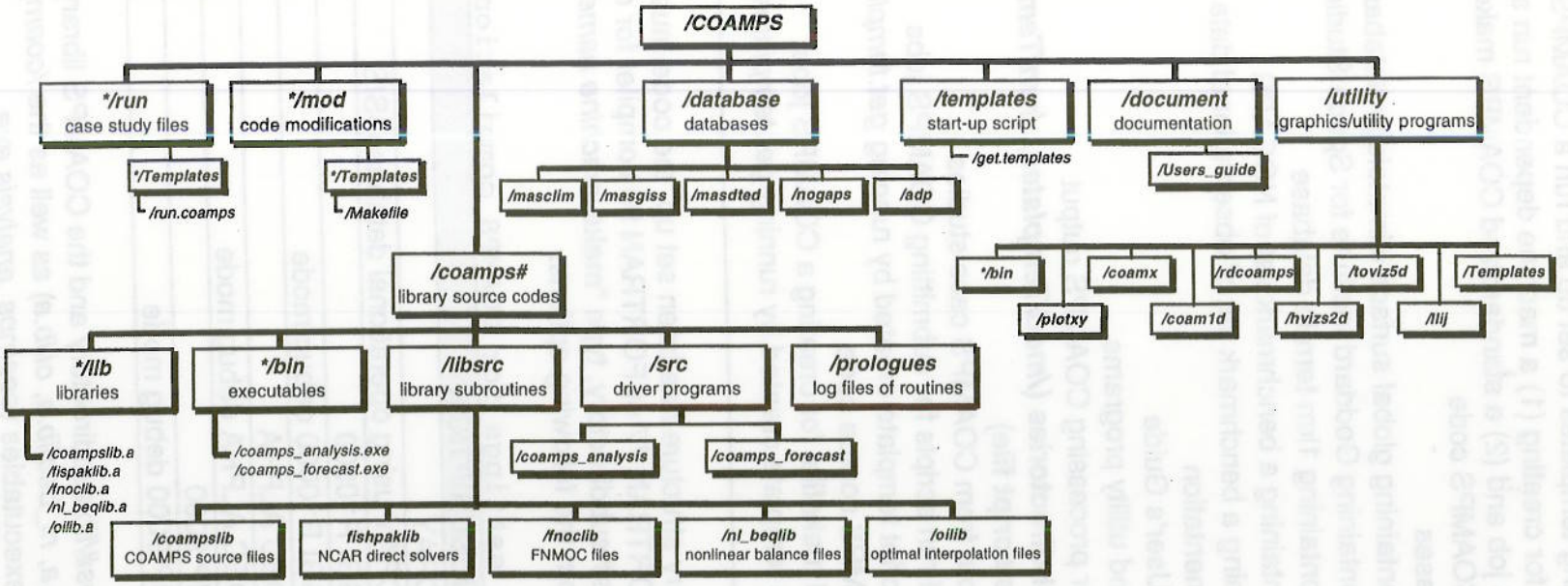


Figure 5: COAMPS directory structure contains five standard subdirectories: */coamps#*, where # is the version release number, */database*, */templates*, */document*, and */utility*. These subdirectories are obtained when the COAMPS tar files are unarchived. The two additional subdirectories, */run* and */mod* are labeled with an asterisk to indicate that they are created later by running the script file *get.templates* in the */templates* subdirectory. Similarly, the */coamps#/lib* and */coamps#/bin* are also created later when the user performs a "**make**" of the master COAMPS code in the */coamps#* subdirectory.

- */templates* - files for generating templates to set up and run a COAMPS job
  /get.templates - script file for creating (1) a machine dependent run script used to submit a COAMPS job and (2) a standardized COAMPS makefile used to compile modified COAMPS code
- */database* - COAMPS databases
  /masclim - subdirectory containing global surface climatology database
  /masgiss - subdirectory containing Goddard Institute for Space Studies database
  /masdted - subdirectory containing 1km terrain database
  /nogaps - subdirectory containing a benchmark set of NOGAPS
  /adp - subdirectory containing a benchmark set of observational data
- */document* - COAMPS documentation
  /Users_guide - COAMPS User's Guide
- */utility* - COAMPS graphics and utility programs
  /Templates - script files for processing COAMPS output
  (The following additional subdirectories (*/mod/Templates, /run/Templates*) are created by the *get.templates* script file)
- */run* - run scripts for and output from COAMPS case studies
  /Templates - standardized run scripts for submitting COAMPS jobs
  /run.coamps - run script template created by running *get.templates*
- */mod* - modifications to COAMPS source code
  /Templates - standardized makefile for creating a COAMPS jobs
  /Makefile - makefile template created by running *get.templates*

Once the COAMPS directory structure has been set up, the code must be compiled. This step requires a FORTRAN 77 or FORTRAN 90 compiler for dynamic storage allocation. In the */coamps#* subdirectory, type "**make** *machine name*". The '*machine name*' can be chosen from the following options:

Table 3: Standard machine settings for COAMPS compilation.

MACHINE NAME	DESCRIPTION
cray	CRAY
cray_with_isis	CRAY using operational database ISIS
sgi	SGI R10000
sgi_g	SGI R10000 debug mode
alpha	DEC ALPHA
alpha_g	DEC ALPHA debug mode
hp	HP 9000
hp_g	HP 9000 debug mode

This command creates the */coamps#/lib* subdirectory and the COAMPS library archives (*coampslib.a, fishpacklib.a, fnoclib.a, nl_beqlib.a, oilib.a*) as well as the */coamps#/bin* subdirectory and the master code executables (*coamps_analysis.exe, coamps_forecast.exe*).

The generation of these COAMPS executables allows the user to proceed to the next step: running a COAMPS job. This is typically accomplished by submitting a COAMPS run script. Obtaining a run script requires running the script file *get.templates* located in the */coamps#/templates* subdirectory. Within this subdirectory, type "**get.templates** *machine name*" inserting the appropriate machine name from Table 3. This command prompts the user for input and upon completion, creates two new subdirectories: subdirectory */coamps#/run/Templates* containing a standardized run script template called *run.coamps*, and subdirectory */coamps#/mod/Templates* containing a standardized makefile template called *Makefile.* This *Makefile* is used to create updated COAMPS executables when the user modifies the original master source code. The procedure for modifying the master COAMPS code and submitting a COAMPS simulation is discussed in Section 4. First, a review of the COAMPS coding framework is given in the next section.

3.3. COAMPS Code

COAMPS consists of two separate driver programs that execute the analysis and the forecast. In the following sections, the flow chart diagrams and primary subroutines associated with each of these two programs are summarized and outlined.

3.3.1. Analysis

The purpose of the analysis routine *coama.F* is to produce a set of initial conditions for the forecast model. In addition to specifying the surface parameters and basic state arrays, the analysis specifies 3D atmospheric fields either by interpolating NOGAPS fields to the COAMPS grids or by blending observations with the first guess fields using the MVOI (multivariate optimum interpolation) analysis. Additionally, *coama.F* prepares lateral boundary conditions for the forecast model.

Figure 6 shows a diagram of the analysis routine's input/output including user-defined namelist values, first guess fields, and surface databases as input, and initial forecast fields as output. Each of the inputs are specified by the user to customize the initial conditions and properly set up a COAMPS simulation. In the analysis code, the *gridnl* namelist parameters are passed in as subroutine arguments. These parameters include: (1) the grid dimensioning variables, which are used later in the code to indicate do-loop limits (**ma(nn),na(nn),kka**), and (2) the array allocation length **nwords** which is used to dimension the total analysis program array **a(nwords)**. Additionally, the pointer location arrays (**ivar (nn)**) are dimensioned in common blocks which are incorporated via the include file *apointers.h*. Then the *coamnl* namelist parameters are read in to specify model features and case study details. This namelist, containing nearly 200 input variables, is the primary vehicle for the user to easily interact with the system without actually making coding changes. The following information is provided to the analysis routine through the *coamnl* namelist inputs:

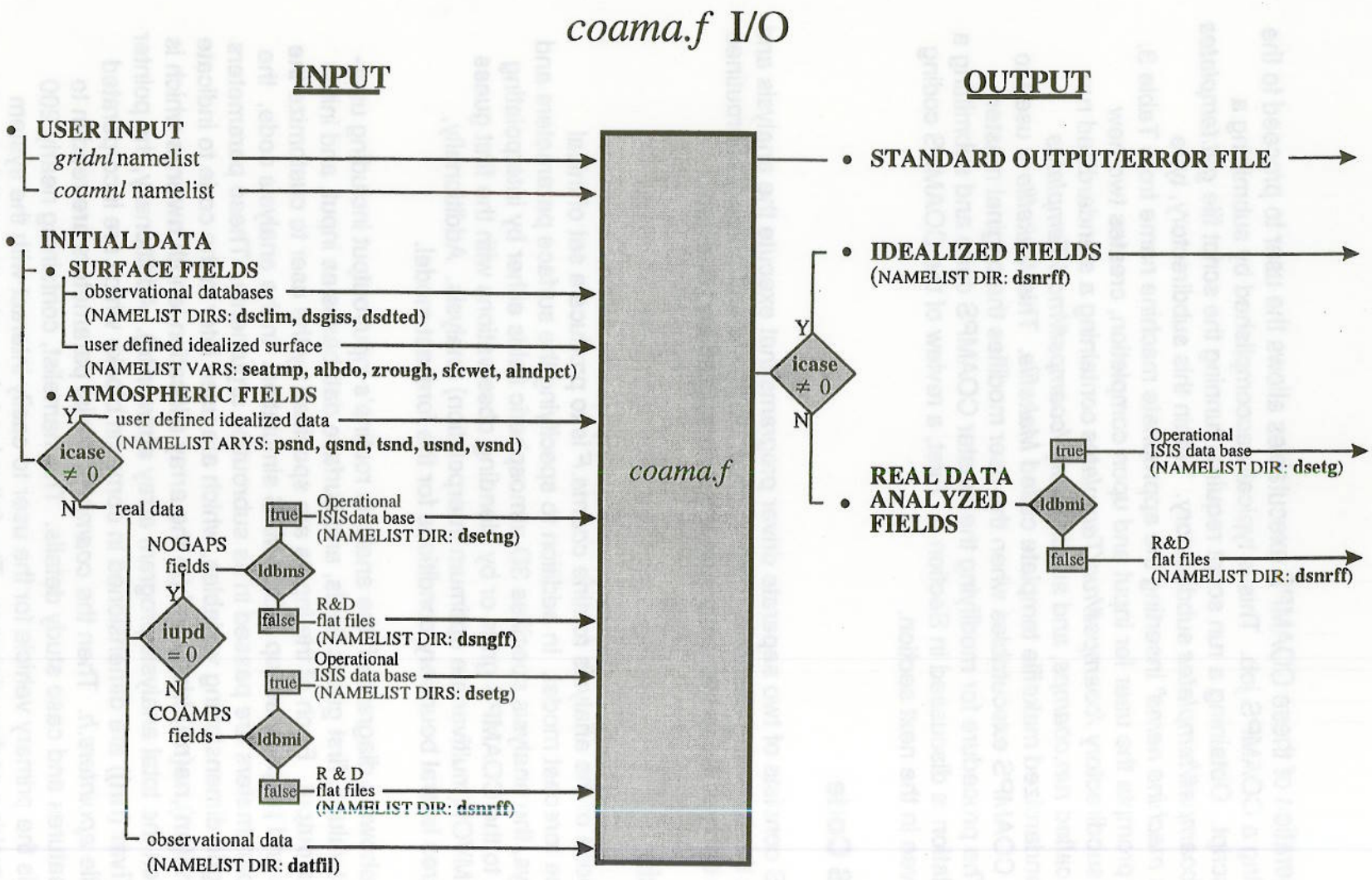


Figure 6: Flow chart of input/output for the analysis routine *coama.f*. The analysis routine's input requirements include user input from the namelists, as well as initial surface and atmospheric fields. The analysis routine outputs a standard output/error file and the idealized or analyzed real data fields for the forecast program. For reference, corresponding namelist parameters are shown in parentheses.

Type of Model Run (Namelist variables)	Directory Locations of Initial Input Data (Namelist variables)
•operational or R&D (**ldbms, ldbmi, ldbmo**) •idealized or real data (**icase**) •data assimilation update (**iupd, loi, loimf**)	•surface databases (**dsclim, dsgiss, dsdted**) •observational data (**datfil**) •NOGAPS fields (**dsetng, dsngff**) •COAMPS fields (**dsetg, dsnrff**)

These parameters, defined further in Appendix D, indicate the location and type of data required to set up the initial conditions. Note that before the analysis can be run, the user must obtain, and specify the directory location of this initial input data. Either idealized or real data may be used to provide the initial fields. Furthermore, real data model runs can be done in either an operational or R&D mode.

Operational use of COAMPS is accomplished by toggling the three database logical variables (**ldbms, ldbmi, ldbmo**) to true, accessing Fleet Numerical Meteorology and Oceanography Center's (FNMOC) operational databases that are managed by the Integrated Stored Information System (ISIS). For R&D purposes however, COAMPS reads in real data initial input fields from 'flat files' using 32-bit IEEE unformatted FORTRAN I/O. In that case, the user must obtain the required data (first guess fields, surface databases, observations and lateral boundary conditions), in the required format (32-bit IEEE unformatted FORTRAN files), and specify the directory location in which each resides (**dsclim, dsgiss, dsdted, datfil, dsetng, dsngff, dsetg, dsnrff**). Once the initial input fields are processed in *coama.F*, they are written out to either '*restart*' files for idealized R&D output, to the ISIS database for operational output, or to flat files for real data R&D output. The flat files are named according to a standardized naming convention given in Section 3.3.2.3 Writing Output, Table 4.

Proper processing of the initial input fields to obtain initial conditions fields requires an understanding of the *coama.F* coding structure. The flow diagram shown in Figure 7 lists the subroutines called by *coama.F*. The first section of the analysis routine consists of reading in the namelist and performing initial setup functions. Once the *coamnl* and *gridnl* namelist are read in, calls to subroutines *chekmn.F, checkz.F, grid.F* and *hm2uv.F* specify the model domain space, map factors and grid nesting.

Then the surface characteristics for each nest are determined from subroutine calls to *sfcpar.F, getsst.F, user_sfc.F,* and *soiltp.F* . In *sfcpar.F*, the surface fields (**rland, albedo, gwet, z0, csst**) are initially read in from a global climatology database. For specific regions, higher resolution data for the **albedo** and **z0,** available from the Goddard Institute for Space Studies (GISS), replaces the climatological values. In subroutine *getsst.F,* the resolution of the surface temperature **tsea** over water is improved by using a 1° NOGAPS field or a 125km x 125km hemispheric grid derived from the Optimum Thermal Interpolation System (OTIS), or from COAMPS history

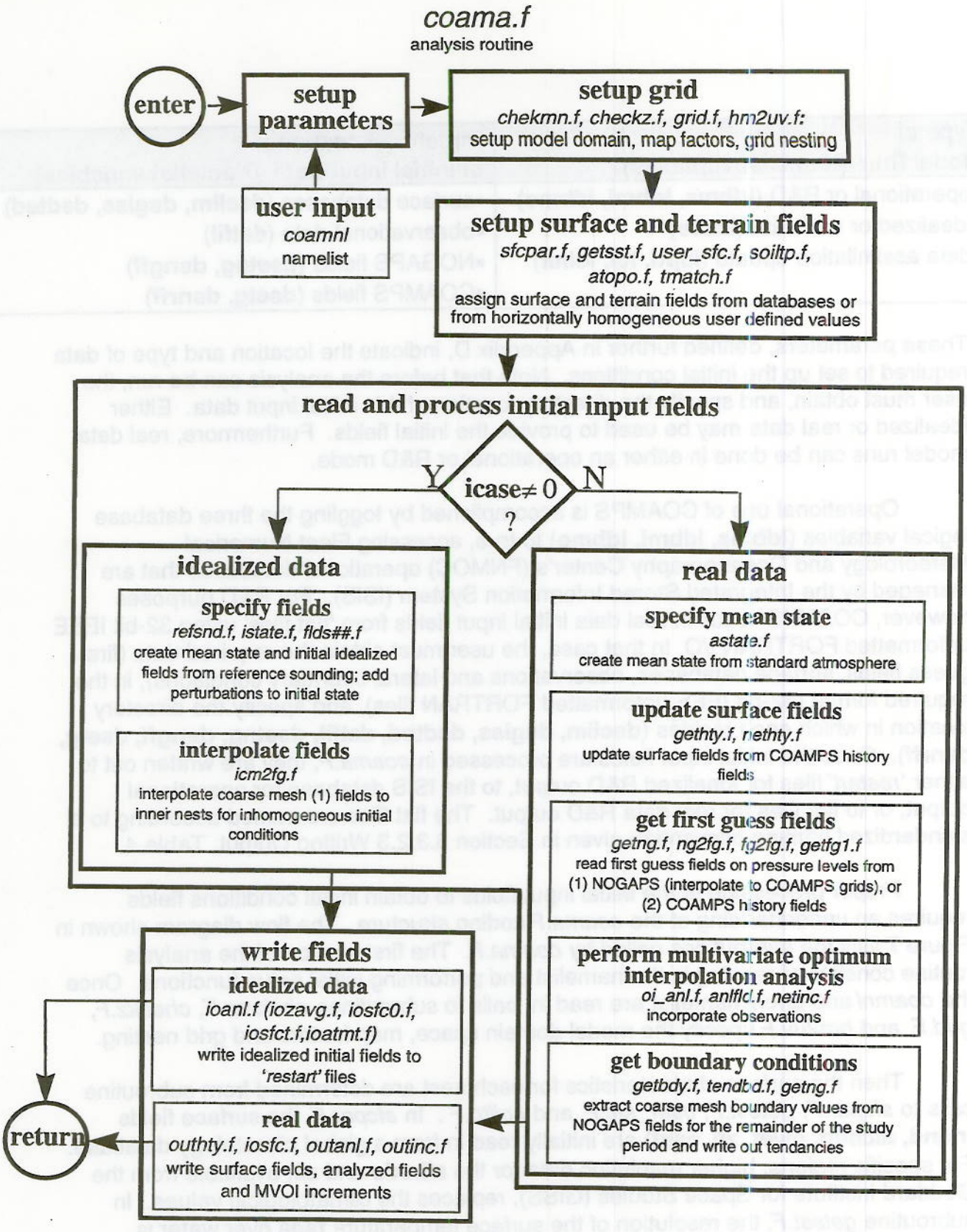


Figure 7: Flow chart of the analysis routine *coama.f*. The analysis routine is described in five parts: setup parameters, setup grid, setup surface and terrain fields, read and process initial input fields and write fields. The subroutines called in each part are shown in italics. The atmospheric fields may be either idealized or real data as indicated by the value of namelist parameter **icase**.

field if either of these source are available. Otherwise, the climagological value **csst** is used. In subroutine *soiltp.F*, values for the deep soil temperature array **tsoil** are generated from climatological data. Also generated in subroutine *sfcpar.F* is the surface topography field (**zsfc**), initialized from a 20km terrain dataset. In certain regions around the globe, 1km resolution terrain data is used for meshes with grid spacing less than 20km.

In setting up the surface characteristics of a particular case study, subroutine *user_sfc.F* allows the user to customize the surface parameters with idealized values through namelist input. For example, user-defined, horizontally homogeneous values for the land/sea table, surface temperature, roughness, albedo and ground wetness arrays are given by namelist parameters **alndpct, seatmp, zrough, albdo, sfcwet** when the namelist flags **ilndflg, iseaflg, iz0flg, ialbflg, igwtflg** are set to '0'. The namelist flag **itopoflg** when set to '0' assigns the terrain height to zero everywhere. Finally, subroutine *tmatch.F* matches the terrain field across the mesh boundaries and subroutine *atopo.F* computes the terrain gradient arrays. The coding then branches for idealized or real data initial conditions as described in the next two sections.

3.3.1.1. Idealized Data

Idealized initial conditions are typically provided by user-defined profiles of winds, temperature, pressure and moisture assumed to be horizontally homogeneous across the model domain. These profiles are read in through namelist arrays **usnd, vsnd, tsnd, psnd, and qsnd** which are used when the namelist parameter **icase** is nonzero. The subroutine *refsnd.F* converts the initial user-defined reference sounding, into MKS units. Then a homogeneous idealized mean state is determined from that data in subroutine *istate.F* for each of the model nests. Perturbations can be added to the mean state by specifying a value for the namelist parameter **icase**='##'. This number '##', corresponds to a subroutine *flds##.F* where the user assigns the field perturbations. When using inhomogeneous idealized fields, set the namelist parameter **jcm2fg** to 1 to interpolate the outer coarse mesh (1) fields to the inner nests. This interpolation is done in subroutine *icm2fg.F*. In subroutine *ioanl.F*, the complete set of idealized initial conditions (surface, basic state and prognostic fields) is written to files that begin with the prefix '*restart*'. These '*restart*' files become the initial forecast fields read in by *coamm.F* to begin an idealized simulation. As discussed later, the '*restart*' files can be created at any time during the COAMPS model run and are also used to initiate COAMPS at a nonzero forecast time (**ktaust**≠0).

3.3.1.2. Real Data

An alternative to idealized initial input fields is the use of global or mesoscale first guess fields that may be updated with observations. The first guess can be obtained from either a NOGAPS analysis or forecast, or from a previous COAMPS forecast valid at the desired date-time group (history fields).

First, the basic state profiles are specified through subroutine *astate.F* . Then the previously specified surface parameters are replaced with higher resolution COAMPS history fields, if they are available. From calls to *gethty.F* and *nethty.F*, the 2D arrays for surface parameters on each nest are updated if the logical namelist parameter **lnrhty**=.true.. These fields (**gwet, z0, tsea**), initially read in from a global climatology database, are assigned values given by the COAMPS history fields. While the array for snow coverage (**snow**) is given values from the NOGAPS global field, the other 2D surface fields obtain values from the COAMPS history fields. These fields include: the 10m winds (**u10m, v10m**), boundary layer depth (**blht**), surface fluxes (**hflxs, hflxl**) and wind stress (**stres**). When **lnrhty**=.false., or if history fields are not found, this latter set of fields are set to zero initially.

The procedure for obtaining the first guess fields begins with subroutine *getng.F* that reads in NOGAPS fields for:

- surface pressure
- u- and v-momentum
- geopotential heights
- vapor pressure

These fields are specified at 1° horizontal resolution and at pressure levels **pr(lm)**, where **lm** and **pr** are user defined namelist parameters defined further in Appendix D. The fields are then interpolated to the coarse mesh (1) grid points in subroutine *ng2fg.F*. Subroutine *fg2fg.F* performs the interpolation of the coarse mesh (1) to the inner nests. If a previous COAMPS forecast is available and the proper namelist flag is set (**iupd**≠0), then subroutine *getfg1.F* is called to overwrite the above fields with COAMPS history fields for each nest. Both routines *getng.F* and *getfg1.F* search backward through seven date-time groups in an attempt to provide COAMPS with the most recent data.

Finally, the first guess fields are adjusted based upon observational data via a multivariate optimum interpolation (MVOI) analysis performed in the subroutine *oi_anl.F* (**loi**=.true.). Note that the MVOI analysis subroutines are not stored with the rest of the COAMPS source code (See Section 3.2, and Figure 5), but are contained in a separate subdirectory and compiled into a separate library. The *oi_anl.F* subroutine determines height and momentum increments based upon differences between the first guess fields on the coarse mesh (1) and ADP observations (i.e. rawinsonde, SSMI, satellite derived data, etc.). These increments are added to the first guess fields and a 9-point smoother is applied in subroutine *anlfld.F*. The same procedure can be used to obtain MVOI increments directly on the inner nest first guess fields (**loimf**=.true.). Alternatively (**loimf**=.false.), the coarse mesh (1) MVOI increments can be interpolated to the inner nest grids and then added to the inner nest first guess fields using subroutine *netinc.F* ; however, this procedure is less effective at maintaining the mesoscale structure of the observations. Using a Cressman scheme, additional analyses are also available on the

pressure levels for temperature (**ltanl**=.true.) and dew point depression (**lqanl**=.true.), as well as for surface temperature (**ltanls**=.true.).

To produce a forecast from real data, boundary conditions for the prognostic fields must be supplied to the coarse mesh (1) at a given time interval (**itauin**). For a specified forecast length (**itauf**), the boundary values are extracted from the NOGAPS forecast fields in subroutine *getbdy.F*, and the tendencies are computed and written out in subroutine *tendbd.F*, again following the standard naming convention.

The final set of real data initial conditions is written out by three output subroutines. Subroutines *outhty.F, iosfc.F* write out the surface fields while subroutine *outanl.F* writes out the analyzed fields at the **pr(lm)** pressure levels. For use in data assimilation, subroutine *outinc.F* writes out the MVOI increments (**loi**=.true.). Since the basic state arrays are easily recomputed, they are not written out by the analysis routine for real data cases. The files created by the analysis are named following the standard convention described in Section 3.3.2.3 for which each filename begins with a four character prefix identifying the field of data it contains. This completes the coding structure associated with the analysis subroutine *coama.F* and the preparation of COAMPS analyzed fields.

3.3.2. Forecast

The purpose of the *coamm.F* code is to produce predicted values of the time-dependent variables. Associated input/output tasks include reading in the initial conditions and writing out the model results. These I/O functions are shown in Figure 8. As in the analysis routine, first the *gridnl* namelist parameters, passed in as subroutine arguments, are used to indicate do-loop limits (**ma(nn),na(nn),kka)**, and the array allocation length **nwords** is used to dimension the total forecast program array **a(nwords)**. Additionally, the pointer location arrays are dimensioned in common blocks which are incorporated via the include file *mpointers.h*. Then the *coamnl* namelist parameters are read in to indicate model features and case study details specified by the user. The 2D graphics instructions, designated here by filenames '*ocards*' and '*xcards*', are read in so that the proper output fields will be saved for graphical display. Finally, the analysis fields created by *coama.F* are input from either the '*restart*' files, ISIS database, or R&D flat files. The forecast routine produces output at specified time intervals for evaluation of model results. COAMPS output includes a standard output/error file, and additional output as requested by the user: '*restart*' files, 1D, 2D and 3D graphics files.

With an understanding of the forecast routines I/O, the *coamm.F* coding is described below. The forecast is produced by integrating a set of model equations governing the prognostic nature of each time-dependent variable. Where necessary to

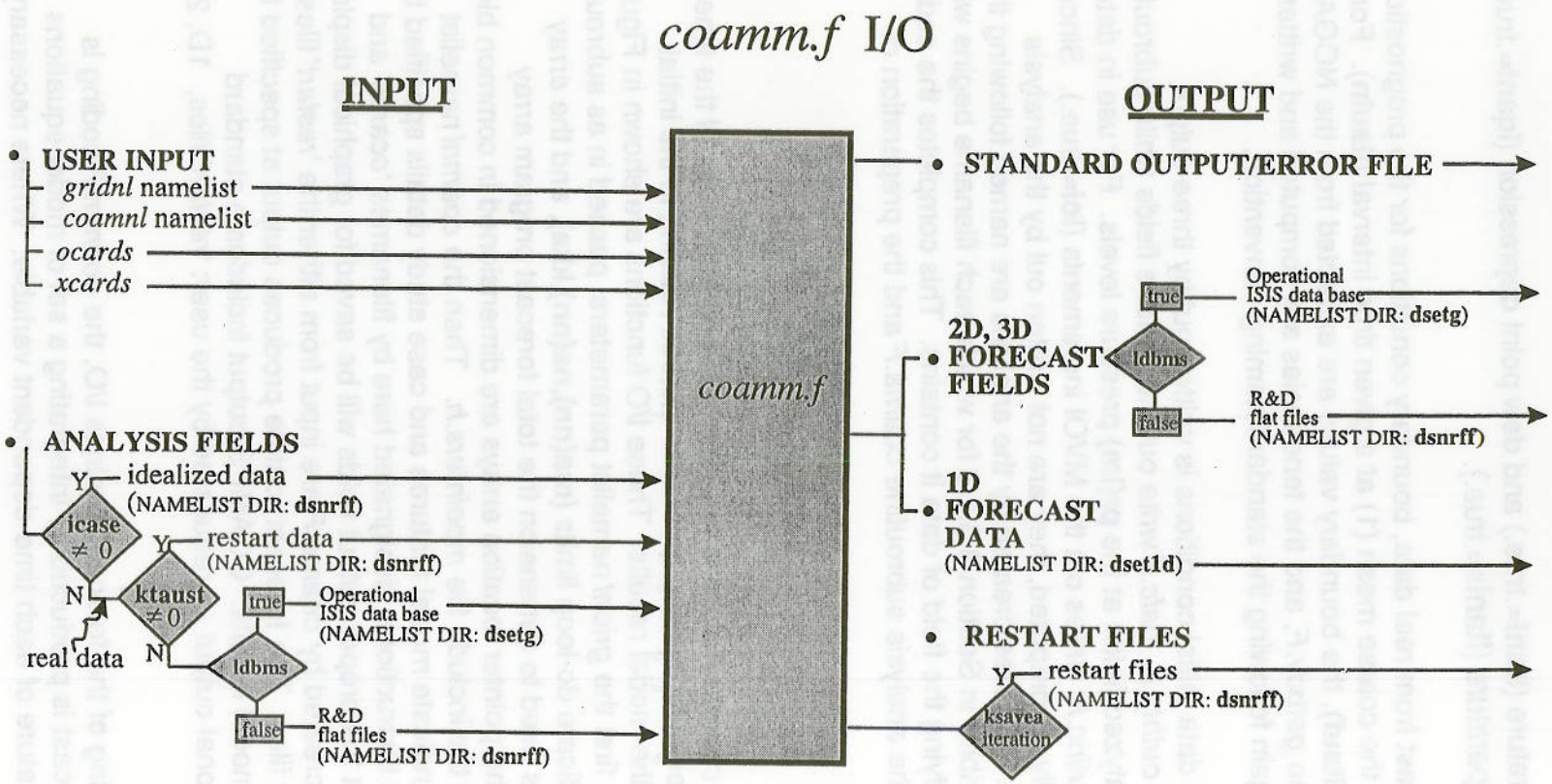


Figure 8: Flow chart of input/output for the forecast routine *coamm.f*. The forecast routine uses from the *coamnl* namelist, input parameters **icase** to setup idealized or real data, **ktaust** to determine if '*restart*' files or if the analysis' first guess fields are input, and **ldbms** to access the operational database or R & D flat files. The *coamnl* namelist also assigns the directory locations, shown in parentheses, of data input/output. The files *ocards* and *xcards* contain user-specified graphics directives. For reference, corresponding namelist parameters are shown in parentheses.

close the system of equations or to represent certain processes, physical parameterizations have been utilized. The following section discusses the coding structure and subroutine calls within subroutine *coamm.F*. Aside from initial setup functions performed by the subroutine, it is subdivided into two other parts: reading in initial conditions and performing the forecast. While reviewing this section, refer to the flow chart and subroutines shown n Figure 9.

3.3.2.1. Reading Initial Conditions

Subroutine *coamm.F* begins by reading in the *coamnl* namelist to allow for user-defined model input and then performs several setup functions that initialize and define constants, parameters, variables and arrays used later in the forecast routine. Additional setup code is embedded in subroutine *coami.F*: the model domain space, map factors and grid nesting are determined from routines *grid.F, hm2uv.F,* and 2D graphics instructions are read in from routines *reado.F, readx.F.* Then, the assignment of COAMPS initial forecast fields is done in *coami.F*. If idealized data (**icase**≠0) are used or a restart run is initiated (**ktaust**≠0), subroutine *iomdl.F* is called to open the '*restart*' files to provide the initial conditions. Otherwise, real data initial conditions are obtained from subroutine calls to *insfc.F, atopo.F, astate.F, inlvl.F.*

When using real data, subroutine *inlvl.F* specifies the initial forecast fields based upon the type of data assimilation select by the namelist parameter **iupd**. When **iupd**=0, a NOGAPS cold start is performed by interpolating analyzed NOGAPS fields on pressure levels to COAMPS sigma levels and horizontal grid points. When **iupd**=1, a full COAMPS update is performed by interpolating the analyzed COAMPS fields on pressure levels to sigma levels (subroutines *instdp.F, stdp2z.F*). When **iupd**=2, an incremental update is performed in subroutine *incrup.F*, where the COAMPS history fields on model sigma levels and the MVOI increments on pressure levels are read in. After the increments are interpolated to the model sigma levels, variational adjustments are made on the pressure and potential temperature increments to bring the fields into hydrostatic balance and then the increments are added to the COAMPS forecast.

Returning back to subroutine *coami.F*, next a few additional preliminary steps are taken before performing the model forecast. Boundary values are extracted in subroutines *readbd.F* and *tendbd.F*, and several model parameters are computed using namelist input. For example, lateral boundaries and weighting functions are determined from **kgetbc, ibdya, jbdya**; the mixing length array is defined using **iamxgl**; the sponge layer is specified by **lralee, lspong, nrdamp**; and the diffusion coefficients are determined from **dif4th** and **dif2nd**. In addition, the prognostic fields are broadcast to the other time levels in *asetup.F*. If the logical variable **linit**=.true., subroutine *ainit.F* is used to modify or update certain fields used by the model. (Each of these user-specified parameters are further defined in Appendix D.) Finally, the initial forecast fields are written out by subroutine *output.F*.

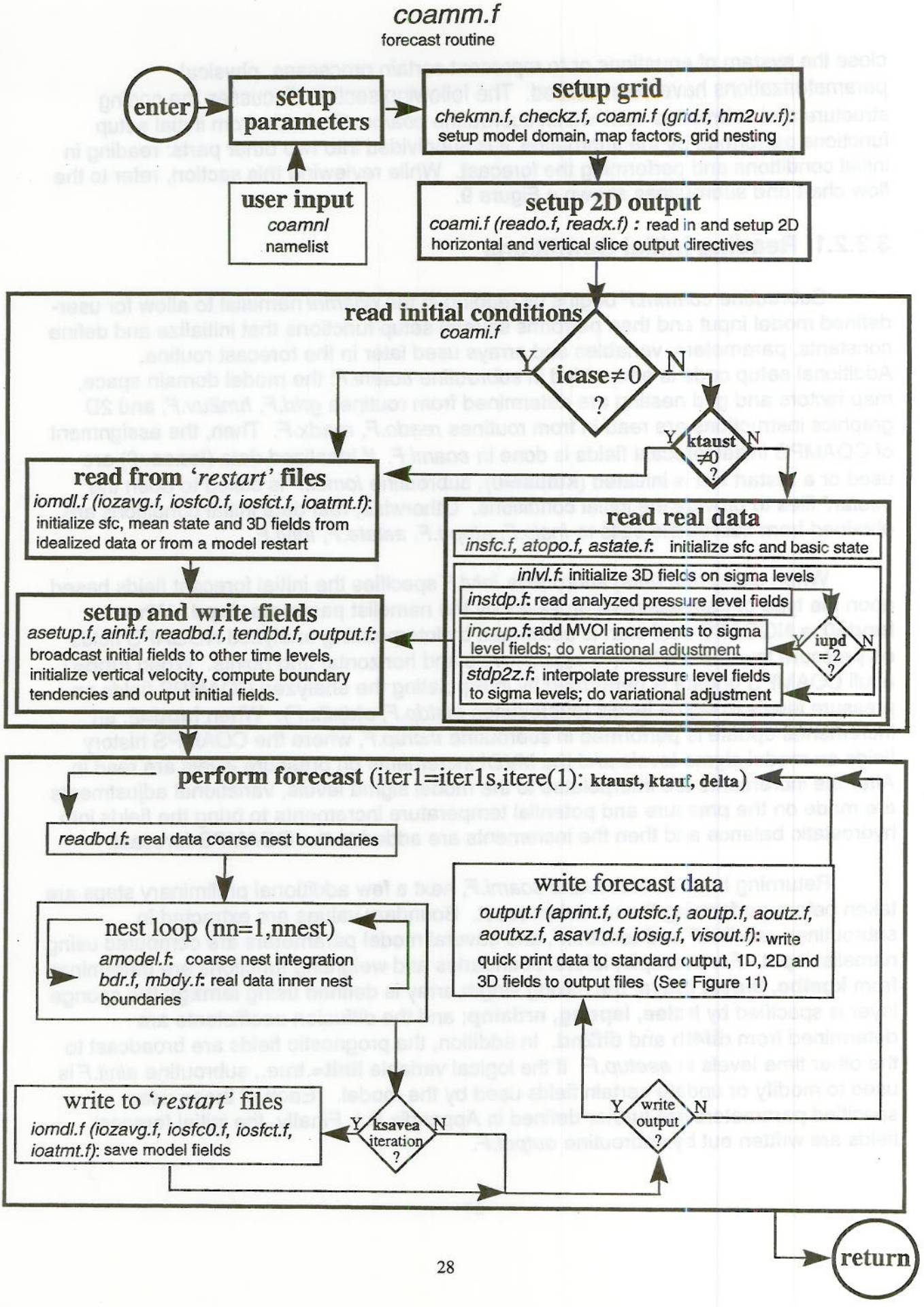

coamm.f
forecast routine
enter
setup parameters
setup grid
chekmn.f, checkz.f, coami.f (grid.f, hm2uv.f): setup model domain, map factors, grid nesting
user input
coamnl
namelist
setup 2D output
coami.f (readx.f, readx.f) : read in and setup 2D horizontal and vertical slice output directives
read initial conditions
coami.f
icase≠0 ?
Y
N
ktaust ≠0 ?
read from 'restart' files
iomdl.f (iozavg.f, iosfc0.f, iosfct.f, ioatmt.f): initialize sfc, mean state and 3D fields from idealized data or from a model restart
read real data
insfc.f, atopo.f, astate.f: initialize sfc and basic state
inlvl.f: initialize 3D fields on sigma levels
instdp.f: read analyzed pressure level fields
incrup.f: add MVOI increments to sigma level fields; do variational adjustment
iupd = 2 ?
stdp2z.f: interpolate pressure level fields to sigma levels; do variational adjustment
setup and write fields
asetup.f, ainit.f, readbd.f, tendbd.f, output.f: broadcast initial fields to other time levels, initialize vertical velocity, compute boundary tendencies and output initial fields.
perform forecast (iter1=iter1s,itere(1): ktaust, ktauf, delta)
readbd.f: real data coarse nest boundaries
nest loop (nn=1,nnest)
amodel.f: coarse nest integration
bdr.f, mbdy.f: real data inner nest boundaries
write forecast data
output.f (aprint.f, outsfc.f, aoutp.f, aoutz.f, aoutxz.f, asav1d.f, iosig.f, visout.f): write quick print data to standard output, 1D, 2D and 3D fields to output files (See Figure 11)
write to 'restart' files
iomdl.f (iozavg.f, iosfc0.f, iosfct.f, ioatmt.f): save model fields
ksavea iteration ?
write output ?
return

Figure 9: Flow chart of the forecast routine *coamm.f*. The forecast routine is described in five parts: setup parameters, setup grid, setup 2D output, read initial conditions and perform forecast. The subroutines called in each part are shown in italics. Depending on the value of two input parameters, **icase** and **ktaust**, *coamm.f* reads data from the '*restart*' files, or reads real data. For real data assimilation, input parameter **iupd** indicates a 'full' or 'incremental' update. Input parameters **ktaust, ktauf, delta** determine the number of coarse mesh iterations (**iter1**). Within this loop, the coarse mesh and inner nest fields are integrated in subroutine *amodel.f* (see Figure 10). Fields necessary to restart COAMPS are saved when indicated by **ksavea**, and various output parameters indicate when forecasted data is written out (see Figure 11).

At this point, control returns to the main forecast subroutine *coamm.F*. And since all model arrays, variables and parameters have been assigned values, COAMPS is now ready to begin integration of the initial forecast fields.

3.3.2.2. Model Integration: Forecast Loop and Atmospheric Model

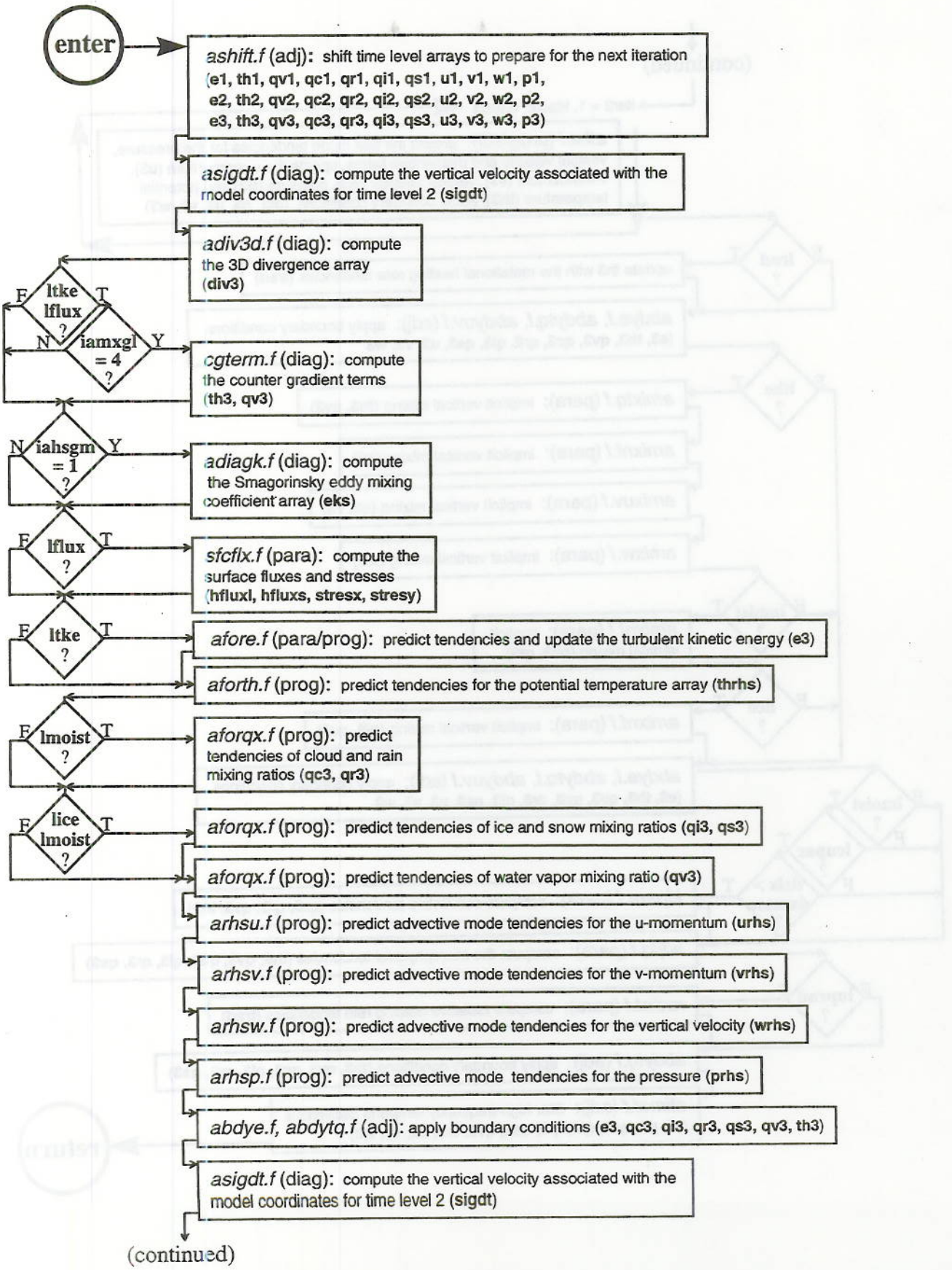
Predicted values of the model fields are obtained by second order leap frog integration of the nonhydrostatic system of equations. These atmospheric equations are coded in subroutine *amodel.F* which is discussed in the next paragraph. The integration of the prognostic fields is controlled by 'the forecast loop' of *coamm.F* (See Fig. 9). The loop indices **iter1s** and **iter1e** are determined from user-specified namelist input **ktaua** and **ktauf.** Each pass through the forecast loop integrates the coarse mesh (1) fields forward one **delta** time step. In real data cases, coarse mesh (1) boundary values are first obtained from subroutine *readbd.F* every **kgetbc** hours. Then the predicted (time level 3) coarse mesh (1) fields are returned by subroutine *amodel.F*. If parameter **nnest**>1, this step is followed by nested calls to *amodel.F* to integrate the fields on each of the inner nests. Because each inner nest is required to be one-third the horizontal grid spacing of it's parent mesh, it is iterated three times for each parent mesh iteration, over a time step that is one-third that of it's parent . At the beginning of each inner nest iteration loop, boundary values are extracted from the parent mesh in subroutine *bdr.F* and *mbndy.F* (for real data cases and only when **nbdya**=7) and the predicted inner nest fields are returned by subroutine *amodel.F*. At the end of each inner nest iteration loop, if namelist parameter **l2way**=.true., the forecasted fields are fed back to the parent mesh in subroutine *feedbk.F* to allow for a two-way interactive grids. Before continuing to the next coarse mesh (1) time step, the forecast time is checked to see if it coincides with times specified by the user for saving '*restart*' files (**ksavea**) or model output (see Figure 11 and Section 3.3.2.3). Of primary importance in the forecast loop is the call to the atmospheric model subroutine *amodel.F* described below.

In general, all namelist inputs, model parameters or arrays used during the model integration are passed through the calling arguments to *amodel.F* rather than through common blocks. This allows for more shared memory usage and dynamic memory allocation within each subroutine. As the subroutines called by *amodel.F* are described, refer to the atmospheric model flow chart in Figure 10. The subroutine calls within *amodel.F* are separated into four categories:

- diagnostics - *asigdt.F, adiv3d.F, cgterm.F, adiagk.F*
- parameterizations - *radiat.F, sfcflx.F, afore.F, kfpara.F, adjtq.F*
- prognostic equations - *afore.F, aforth.F, arhsu.F, arhsv.F, arhsw.F, arhsp.F, alhs.F*
- adjustments - *ashift.F, abdye.F, abdytq.F, abdyuv.F, abdyp.F, amixtq.F, amixnf.F, amixuv.F, amixw.F, atimef.F*

amodel.f

atmospheric model equations and parameterizations



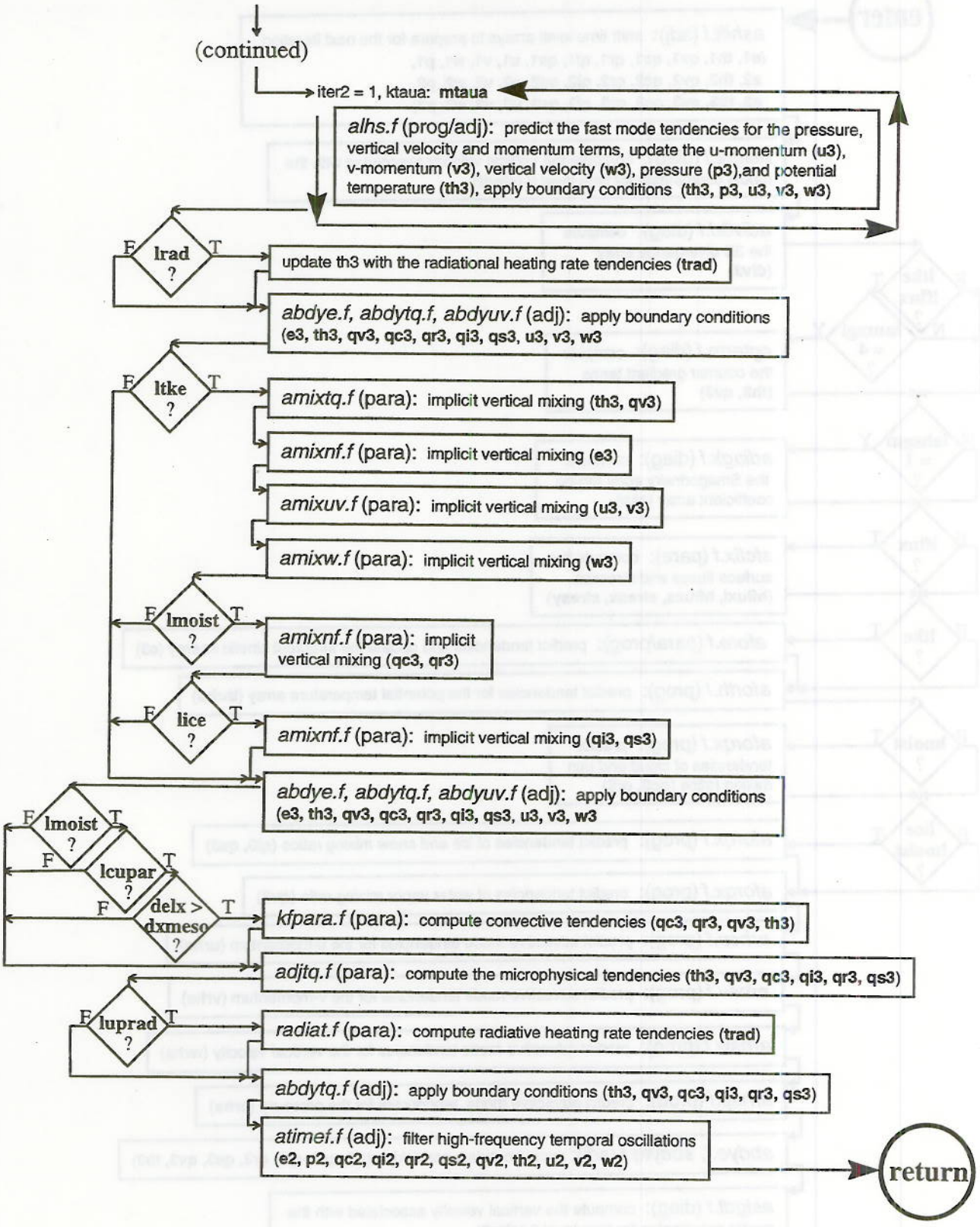

(continued)
iter2 = 1, ktaua: mtaua
alhs.f (prog/adj): predict the fast mode tendencies for the pressure, vertical velocity and momentum terms, update the u-momentum (u3), v-momentum (v3), vertical velocity (w3), pressure (p3),and potential temperature (th3), apply boundary conditions (th3, p3, u3, v3, w3)
lrad ?
F
T
update th3 with the radiational heating rate tendencies (trad)
abdye.f, abdytq.f, abdyuv.f (adj): apply boundary conditions (e3, th3, qv3, qc3, qr3, qi3, qs3, u3, v3, w3
ltke ?
amixtq.f (para): implicit vertical mixing (th3, qv3)
amixnf.f (para): implicit vertical mixing (e3)
amixuv.f (para): implicit vertical mixing (u3, v3)
amixw.f (para): implicit vertical mixing (w3)
lmoist ?
amixnf.f (para): implicit vertical mixing (qc3, qr3)
lice ?
amixnf.f (para): implicit vertical mixing (qi3, qs3)
abdye.f, abdytq.f, abdyuv.f (adj): apply boundary conditions (e3, th3, qv3, qc3, qr3, qi3, qs3, u3, v3, w3
lmoist ?
lcupar ?
delx > dxmeso ?
kfpara.f (para): compute convective tendencies (qc3, qr3, qv3, th3)
adjtq.f (para): compute the microphysical tendencies (th3, qv3, qc3, qi3, qr3, qs3)
luprad ?
radiat.f (para): compute radiative heating rate tendencies (trad)
abdytq.f (adj): apply boundary conditions (th3, qv3, qc3, qi3, qr3, qs3)
atimef.f (adj): filter high-frequency temporal oscillations (e2, p2, qc2, qi2, qr2, qs2, qv2, th2, u2, v2, w2)
return

Figure 10: Flow chart of the atmospheric model subroutine *amodel.f*. Subroutines called by *amodel.f* are designated as: adjustment (adj), diagnostic (diag), prognostic (prog), and/or parameterization (para) routines. Shown in parentheses are the arrays updated in each subroutinecalled by *amodel.f*. The number associated with each array indicates the time level. Many of the model parameterization features are controlled by input parameters. For example, subgrid scale mixing is controlled by parameters **ltke, iamxgl, iahsgm,** surface fluxes by **lflux,** thermodynamic processes by **lmoist, lice,** convective parameterization by **lcupar, dxmeso,** and radiative affects by **lrad** and the internal logical parameter **luprad** which obtains a value based upon the values of both **lrad** and **dtrad**. See Appendix D for a more thorough description of these, and the other COAMPS namelist input parameters.

Within *amodel.F*, first the time levels associated with the prognostic arrays are shifted in subroutine *ashift.F* to prepare them for the next model iteration. To illustrate this process, we use the u-momentum component array as an example:

u1(m,n,kk) = u2(m,n,kk)
u2(m,n,kk) = u3(m,n,kk)
u3(m,n,kk) = u1(m,n,kk).

After *ashift.F,* time level 1 arrays contain previous values (time=t-Δt), time level 2 arrays contain present values (time=t), and by the end of *amodel.F,* time level 3 arrays contain predicted values (time=t+Δt). Note that the array dimensions used in *amodel.F* are more general (**m,n,kk**) since they can represent the dimensions of any mesh.

Before the prognostic tendencies are computed, several preliminary diagnostic routines are called to define arrays used later. These routines include: *asigdt.F* which computes sigma-coordinate vertical velocities (**sigdt**), *adiv3d.F* which computes the two or three-dimensional divergence (**div3),** *cgterm.F* which computes and adds in counter gradient flux terms to the potential temperature and water vapor mixing ratio when **iamxgl**=4 or 5, and *adiagk.F* which computes Smagorinsky type horizontal eddy mixing coefficients when **iahsgm**=1 (**eks**). The surface parametertization routine, *sfcflx.F*, is also called in advance to produce values for the surface fluxes (**hfluxl, hfluxs**) and stresses (**stresx, stresy**) used as lower boundary conditions for the water vapor, potential temperature and wind components respectively.

Next, the predicted future values of the time dependent variables are obtained. The terms associated with the slow or advective modes in COAMPS are computed first in subroutines *afore.F* (**e3**), *aforth.F* (**thrhs**), *aforqx.F* (**qc3, qi3, qr3, qs3, qv3**), *arhsu.F* (**urhs**), *arhsv.F* (**vrhs**), *arhsw.F* (**wrhs**), and *arhsp.F* (**prhs**). In the above routines, time level 3 arrays for TKE and moisture are updated because they do not contain terms related to the faster moving sound and gravity waves which are computed separately. The forcing associated with the fast modes are integrated in subroutine *alhs.F* using a time-splitting and semi-implicit computation over a shorter time step determined from namelist variable **mtaua**. Here the time level 3 arrays for the potential temperature, wind components and pressure are updated adding in the previously computed advective mode tendencies (**thrhs, urhs, vrhs, wrhs** and **prhs**).

Once the time level 3 arrays have been updated, adjustments are made to the prognostic variables. These adjustments include explicit moist physics, radiation, turbulent vertical mixing, and temporal filtering. For example, subroutine *kfpara.F* comprises a convection parameterization scheme used when the logical variable **lcupar**=.true. and when the horizontal resolution of the nest is greater than that specified by namelist parameter **dxmeso.** Subroutine *adjtq.F* is a cloud microphysical parameterization scheme that handles subgrid scale moisture processes when **lmoist**=.true.. Subroutine *radiat.F* parameterizes the long and short wave radiation

effects upon the potential temperature when logical variable **lrad**=.true.. Also listed as a parameterization subroutine is the TKE prediction routine *afore.F* called when **ltke**=.true.. This routine consists of 1.5 order boundary layer closure with several options for parameterizing the eddy mixing coefficients and the turbulent mixing length.

Further adjustments are made to the time level 3 arrays by applying boundary conditions, in subroutines *abdye.F, abdytq.F, abdyt.F, abdyuv.F, abdyw.F, abdyp.F*; and performing vertical mixing, in subroutines *amixtq.F, amixnf.F, amixuv.F, amixw.F* (**ltke**=.true.). Finally, temporal oscillations associated with the leap frog integration scheme are smoothed in subroutine *atimef.F*. To maintain numerical stability, the damping, diffusion, filtering and horizontal mixing computations are performed on the time level 1 arrays, and vertical mixing is done implicitly on the time level 3 arrays. All other quantities are computed on time level 2 arrays. See Section 2 for a list of references that describe the parameterization and adjustment schemes used in COAMPS, and Appendix D for further descriptions of the namelist input parameters.

The coding in the above subroutines completes one model iteration with updated predicted values stored in the time level 3 arrays. Control returns to the forecast routine where the atmospheric model is called for up to six additional inner nests, thus repeating the *amodel.F* coding sequence. At the end of each coarse mesh (1) iteration, *coamm.F* queries whether the forecast fields are written to '*restart*' files (in subroutine *iomdl.F*) and/or graphical display files (in subroutine *output.F*) as described in the next section.

3.3.2.3. Writing Output

COAMPS produces several forms of model output in subroutine *output.F* for viewing case study results and testing code development. Output options include: 2D horizontal and vertical slices of data viewed in numerical format, called 'quick prints', as well as 1D, 2D, and 3D data post-processed through a variety of separate graphics programs. Figure 11 depicts the flow diagram for writing out COAMPS results. Here, the subroutines are briefly introduced beginning with an overview of 2D data output. More detailed information concerning COAMPS graphics is presented in a separate document.

In subroutine *aprint.F*, a predetermined set of model fields is written to standard output in 2D numerical quick print format at time intervals specified by the user (**kprnta**). The standard output file also contains standard error messages, and thus, in addition to numerical output of model fields, also indicates if the job has completed successfully. The 2D horizontal and vertical slice data, generated for graphical display purposes, are specified by the user before a COAMPS model run. This information is read into COAMPS through graphics 'directives' given in files whose names correspond to the input parameters **npfil** (horizontal slice data) and **xsfil** (vertical slice data).

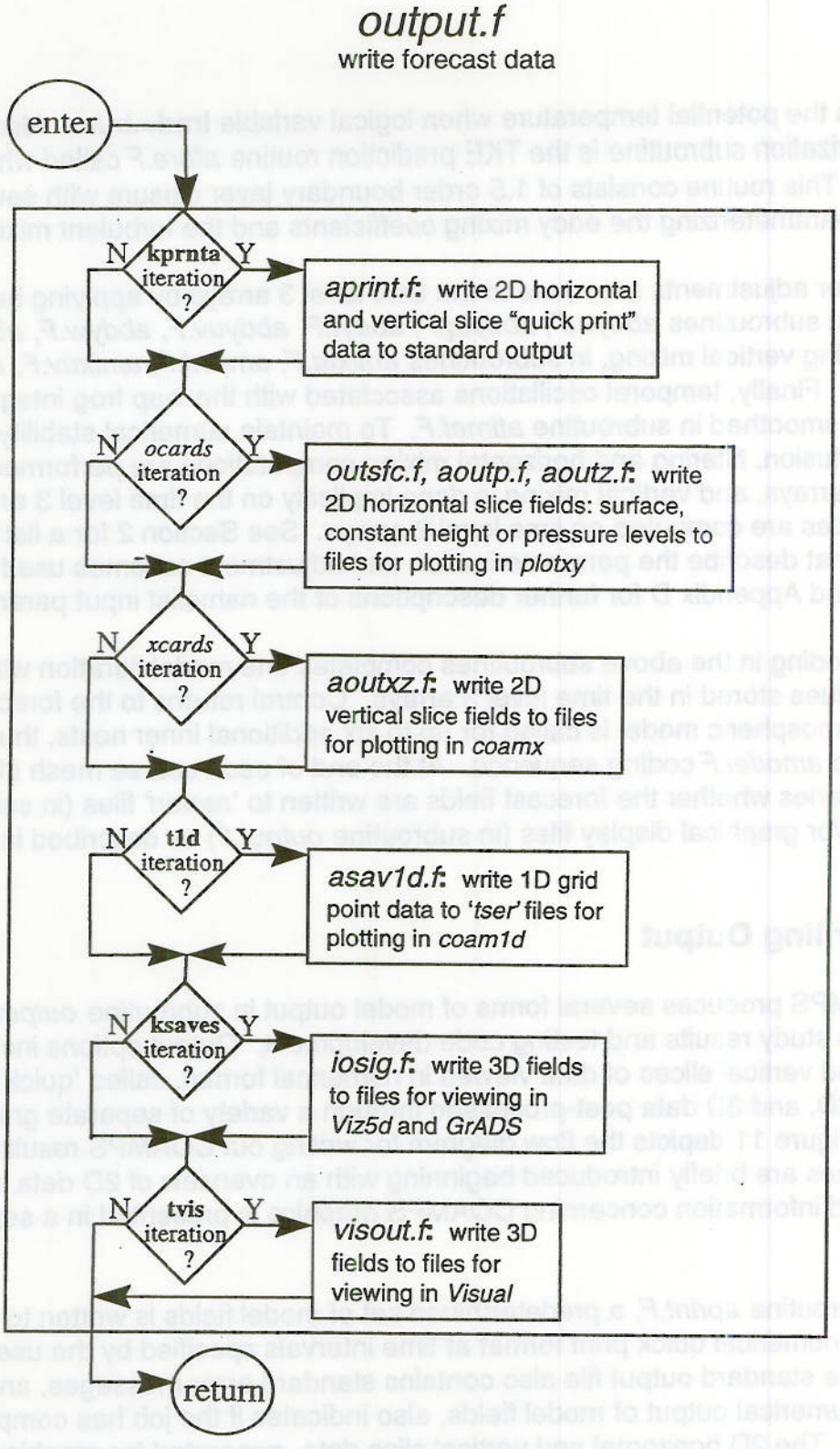
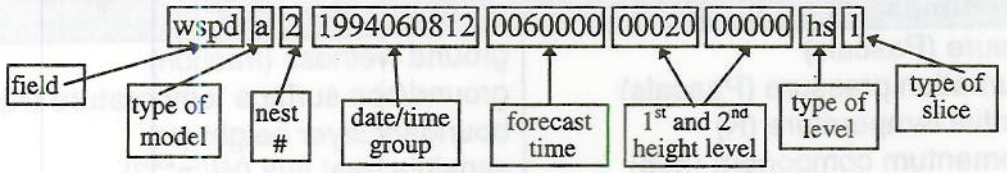


Figure 11: Flow chart of the subroutines called by *coamm.f* that write out model data. The input parameters **kprnta, t1d, ksaves, tvis** represent forecast times, defined in Appendix D, that are converted into iteration numbers,. The *ocards* and *xcards* iterations refer to times given in the 2D graphics directive files created by the user.

The 2D graphics directives are used in subroutines *outsfc.F, aoutp.F, aoutz.F,* and *aoutxz.F,* to produce the following types of output:

- *outsfc.F* - surface fields
- *aoutp.F* - horizontal slices of fields at pressure levels
- *aoutz.F* - horizontal slices of fields at height levels
- *aoutxz.F* - vertical slices of fields

The output files generated are named following a standardized naming convention. Each unique filename uses 36 characters to identify 11 pieces of information about the data it contains. For example, filename *wspda2199406081200600000002000000hsl* represents:



The filename indicates that the data are of horizontal wind speed from the atmospheric model on the second nest. Furthermore, the data are a horizontal slice on a height surface at a height of 20 m, 6 hours into a forecast initialized at 12Z on June 8th, 1994. Table 4 summarizes the information given by the output filename.

Table 4: COAMPS Output File Naming Convention

OUTPUT FILENAME	TYPE	POSITION	OPTIONS-DESCRIPTION
field name indicator	a4	1-4	see COAMPS graphics document
model type indicator	a1	5	a - atmosphere o - ocean
nest number indicator	i1.1	6	1 - coarse mesh 2,3,4,5,6 or 7 - inner nests
date-time group	a10	7-16	YYYYMMDDHH
forecast time: hr min sec	i3.3 i2.2 i2.2	17-19 20-21 22-23	a time within the forecast period evenly divisible by the time step
level: meters	i5.5	24-28	height level (horizontal slice data)
second level: meters	i5.5	29-33	2nd height level (for thickness fields)
level type indicator	a2	34-35	sl - mean sea level hs - height surface ml - marine level sf - surface pr - pressure level
plot type indicator	a1	36	l - horizontal slice x - vertical slice

Subroutines for producing 1D (grid point) and 3D volumes of model data, are briefly mentioned here. The 1D data output is determined from the *coamnl* namelist variables **t1d, dset1d, cpar, city, i1d** and **j1d**. Subroutine *asav1d.F* creates 1D graphics files (with filename prefix '*tser*') at time intervals given by **t1d**. And subroutine *iosig.F* creates 3D volumes of sigma level data (with the filename ending in '*sgl*') and 2D surface data (with the filename ending in '*sfl*') at time intervals given by two namelist parameters: **icycle** (for use in data assimilation) and **ksaves** (for graphical display: *Viz5d, GrADS*). The standard set of output produced by this subroutine is listed in Table 5 for reference.

Table 5: Sigma level data and surface fields written out by COAMPS

3D Volumes	2D Surface Fields
pressure (Pascals)	ground wetness (fraction)
perturbation pressure (Pascals)	ground/sea surface temperature (K)
potential temperature (K)	boundary layer height (m)
u-momentum component (m/s)	sensible heat flux (W/m**2)
v-momentum component (m/s)	latent heat flux (W/m**2)
water vapor mixing ratio (kg/kg)	wind stress (Newton/m**2)
cloud water mixing ratio (kg/kg)	
ice crystal mixing ratio (kg/kg)	
rain drop mixing ratio (kg/kg)	
snow flake mixing ratio (kg/kg)	
turbulent kinetic energy (m**2/s**2)	
vertical velocity (m/s)	

Another option is available for viewing primarily 2D model results using the graphics package *Visual*. Subroutine *visout.F* creates *Visual* files (with the filename ending in '*GRD1*', *GRD2*', or *GRD3*') at time intervals given by namelist inputs **tvis1, tvis2**.

This completes the overview of the COAMPS code. At this point, the flow structure of the code, the subroutine functions, and the use of certain namelist parameters have been described. In the next section, the procedure for submitting a COAMPS job is reviewed.

4. Execution Procedure

The execution procedure for a COAMPS job involves two phases: case study setup and submitting a COAMPS job. Before a COAMPS job begins, several preliminary steps are taken involving preparation of the initial conditions and the code. The steps outlined below show how to get started using COAMPS as a new user, and how to set up a new case study. First time users of COAMPS, need to obtain the COAMPS tar files, set up the COAMPS directory structure and compile the master COAMPS libraries and executables. These steps are covered in Section 3.2. Also described in Section 3.2 are the steps for obtaining two templates: a run script used to run the COAMPS executables, *run.coamps,* and a makefile used to create modified executables, *Makefile.* The steps given here describe how to run a user-customized COAMPS simulation with the option of modifying the original source code.

4.1. Preparation Steps

CODE MODIFICATIONS

(To run unmodified source code, goto step 10.)

1. cd to the subdirectory */coamps/mod*
2. type "**mkdir** *modsd*" where *modsd* is any user-defined subdirectory name that will contain user generated code modifications
3. cd to this new subdirectory *modsd*
4. type "**cp ../Templates/Makefile .**" to get the *Makefile* template into the '*modsd*' subdirectory.
5. copy subroutines from any of the subdirectories in */coamps#/libsrc* for modification
6. modify and add subroutines
7. copy all include files (**.h*) from the subdirectories in */coamps#/libsrc* that are called by the subroutines that have been modified
8. edit the *Makefile* to include object files corresponding to subroutines that have been modified or added
9. make the new executable: type "**make** *machine name*" (See Table 3)

INITIAL INPUT FIELDS

10. obtain the initial input fields
    - <real data case>

      To perform a NOGAPS 'cold start', COAMPS uses 1° resolution NOGAPS fields in flat file format (32-bit, IEEE) and ADP observational data in FGGE format. These data are available from FNMOC and can be obtained using an anonymous ftp. For a typical model run, a NOGAPS analysis or forecast provides the first guess fields and 12 hourly NOGAPS forecasts provide boundary conditions throughout the duration of the simulation study period. In the absence of NOGAPS forecasts, fixed boundary conditions are used (although this is not recommended). Once these data are obtained, they should be copied into the appropriate subdirectory:

/coamps#/database/nogaps and */coamps#/database/adp*. Other databases used in COAMPS (surface fields, terrain characteristics and climatology data) are provided with the initial COAMPS tar file *database.tar*. These data are also required to be in flat file format. (Note that the subdirectory locations of this initial input data must be provided as namelist input: parameters **dsngff, dsclim, dsdted, dsgiss**.)

- <idealized case>
  Obtain or create an idealized sounding (pressure, temperature, moisture and winds) and input these data through namelist arrays **psnd, tsnd, qsnd usnd, vsnd**. Perturbations, used to modify these initial idealized profiles, may be defined by the user in subroutine *flds##.F* . Perturbations are added when the namelist parameter **icase**=## coincides with the number in the *flds##.F* filename. The surface fields can be defined from either user-specified namelist parameters: **alndpct, seatmp, zrough, albdo, sfcwet** or from the standard COAMPS surface databases located in subdirectory */coamps#/database*. (Note that if real surface data is used, the subdirectory locations of the surface databases must be provided as namelist input: parameters **dsclim, dsdted, dsgiss**.)

CASE STUDY

11. cd to the subdirectory */coamps/run*
12. type "'**mkdir** *runsd*' where *runsd* is any user-defined directory name that will contain a case study run script and output files
13. cd to this new subdirectory *runsd*
14. type '**cp ../Templates/run.coamps .**' to get the COAMPS run script template into the *runsd* subdirectory.
15. edit the *run.coamps* run script to customize the namelist input for a particular case study (i.e. modify the *gridnl, coamnl,* and *mvoinl* namelists (See Section 4.2))
16. if the COAMPS executables have been modified, make sure that the **binDir** variable in the *run.coamps* script points to the location of the new executables.
17. create a */coamps/data* subdirectory for graphical and model output data

These steps summarize the preparation phase of a COAMPS job. Correct specification of user input (step 15) is of primary importance in the above sequence. Because COAMPS includes a variety of important features and options as namelist input, the user has the ability to interact with the code through these namelists. Although this is of great benefit, high levels of user input also creates the opportunity for error. To educate the user and reduce the risk of error, categories of user input are grouped below based upon their function with the COAMPS system.

4.2. COAMPS Input File: *Gridnl, Coamnl* and *Mvoinl* Namelists

Perhaps the single most important step in running COAMPS successfully is indicating the proper namelist input. Many of the more important model parameters are specified by the user in the three model namelists *gridnl, coamnl* and *mvoinl*. The parameters from these namelist and their user-specified values are assigned in a single input file which, for purposes of discussion, is called *nlist.input*. This input file is created when the COAMPS run script is submitted (See Appendix B for a sample input file). A complete list of the namelist input parameters, default values, and definitions is given in Appendices C and D arranged alphabetically according to the type of input (i.e. integer, real, character, logical). Categories of user inputs described here include the following parameter groups: model setup, case study setup, and data manipulation.

4.2.1. Model Setup

This section describes the inputs that allow the user to set up the model domain and grid structure, the integration procedure, and the boundary conditions for a particular case study.

4.2.1.1. Grid

Specification of the namelist parameters associated with the grid setup are critical in producing a proper model run. Their values vary from case to case depending on the region/domain of interest, and the vertical/horizontal spacing required to resolve meteorological features of interest.

Input parameters:

the number of nests **nnest**; horizontal grid spacing of the coarse mesh **delx, dely**; vertical grid spacing **dsigma**; number of grid points for each nest **ma(nn), na(nn), kka**; map projection **nproj**; standard latitude(s) and longitude **phnt1, phnt2, alnnt**; reference latitude and longitude **rlat, rlong**; reference grid points **iref(1), jref(1)**; reference grid points of inner nests **ii(nn), jj(nn)** (where **nn** is the nest number)

It is possible to determining the model domain in advance of a COAMPS job by running the script file *domain.sh* located in the */utility/Templates* subdirectory. Edit this file, inputting values for the grid setup parameters, and then type: "**domain.sh**". A meta file of the nested model domain is created and displayed on the screen, provided that the DISPLAY environmental variable has been specified and the 2D graphics executable *plotxy* is available. Once you are satisfied with the model domain, insert those parameter values into the *coamnl* namelist. Examples of three model domains are given in Appendix E. Tables E-1,E-2,E-3 give the namelist parameter values used to set up a model grid based upon the mercator, lambert conformal, and polar stereographic map projections respectively. Figures E-2,E-4,E-6 show the

corresponding COAMPS model domains obtained by running *domain.sh* script for each of the three examples.

4.2.1.2. Integration

COAMPS uses second-order leapfrog time differencing to integrate the slow modes (advective solution) using a time step, **delta**, that depends upon the grid resolution and advective speeds to ensure computational stability. Because COAMPS is nonhydrostatic, the simulations also contain sound and gravity wave solutions. Their faster propagation rates require integration over a smaller forward time step, designated by **delta/mtaua** where **mtaua** is the number of small time steps per large time step. Two important numerical modeling improvements allow for larger values of the small time step. These include semi-implicit vertical differencing and the quasi-compressible assumption which permits the speed of sound to be much less than its typical atmospheric value of ~350 m/s.

Input parameters:

the starting and ending forecast times **ktaust, ktauf**; the large time step **delta**; the number of small time steps per large time steps **mtaua**; and the speed of sound **cbar**

4.2.1.3. Boundary Conditions

These parameters allow for specification of the lateral boundaries and damping at the upper boundary. Five types of lateral boundary conditions are available for idealized simulations and two types are available for real data simulations each defined in Appendix D. Real data boundaries can be computed by the Perkey-Krietzberg method (**ibdya=jbdya=nbdya**=6), which specifies the boundaries according to the NOGAPS tendencies, or by the Davies method (**ibdya=jbdya=nbdya**=7), which uses the actual NOGAPS fields. In general, the Davies method is recommended for most real data model runs. Additionally, two types of upper level sponge layers have also been developed to reduce the amplification of vertical wave reflections off the model top.

Input parameters:

damping upper layer **lralee, lspong, nrdamp, rdtime**; lateral boundary condition indicators **ibdya, jbdya, nbdya**; gravity wave speed (for idealized radiative horizontal boundary conditions) **cstar**; number of grid points in real data lateral boundary computations **nbdypt**; times for obtaining coarse mesh (1) lateral boundary tendencies from NOGAPS fields **itauin, itauf**; time interval for updating coarse mesh (1) lateral boundaries during COAMPS forecast **kgetbc**

4.2.2. Case Study Setup

These parameters govern the model features that are active for a particular case study. They are grouped into three categories: analysis fields parameters, model physics parameters and numerical damping parameters. This group of inputs determine the following characteristics of the case study: type of model run (idealized or real data run), data assimilation features (type of update, MVOI analysis), physical parameterizations active (TKE, radiation, moisture, surface fluxes, surface energy budget), numerical damping options (diffusion, time filter, divergence damping, upper level sponge layers), in addition to miscellaneous dynamical features (advection, geostrophic wind). Note that these inputs can lead to complications when poorly specified.

4.2.2.1. Analysis Fields

These parameters allow for the specification of fields required to run the model. Idealized initial conditions are more simply specified than are real data initial conditions because they are set up by the user in the COAMPS input file. Perturbations from a user generated reference sounding are fed into the COAMPS analysis routine (*coama.F*) through a *flds##.F* file. In contrast, real data setup is considerably more complex using a variety of sources to prepare the initial conditions including NOGAPS analysis or forecasts, COAMPS history fields, climatology fields, surface parameters, and observational data.

Input parameters:

For idealized model runs: the case study indicator **icase**≠0; latitude of f-plane **flat**; interpolation flag for nested fields **jcm2fg**; idealized surface conditions **ilndflg, iseaflg, iz0flg, ialbflg, igwtflg, alndpct, seatmp, zrough, albdo, sfcwet**, **itopoflg**; idealized reference sounding **umean, vmean, ipsnd, itsnd, iqsnd, iuvsnd, psnd(k), tsnd(k), qsnd(k), usnd(k), vsnd(k)** (where **k** is the vertical level index);
For real data model runs: the case study indicator **icase**=0; type of update **iupd**; multi-variate optimum interpolation analysis (MVOI) **loi, loimf**; nonlinear variational balancing **ivar**; directory locations of initial input **dsngff, dsnrff, datfil, dsclim, dsdted, dsgiss**; real data surface conditions **lnrhty, ltopoa, nftopo, gwetmx, z0z0mx**; data assimilation **icycle, itauin, itauf, kgetbc, loi, iupd**: Cressman analyses **ltanl, lqanl, ltanls**

Note that real data surface conditions may also be used for idealized model runs, and idealized surface conditions may be used for real data runs.

4.2.2.2. Model Physics

COAMPS contains many inputs that govern various parameterization options for specifying the surface and boundary layers, moist physics, convection and radiation schemes. Refer to Appendix D for further definitions.

Input parameters:

surface layer processes **lflux, lsfcen, lles**; boundary layer processes **ltke, iahsgm, iamxgl, iashsm, sh0, sm0, lles**; moist and convective processes **ilsp, lmoist, lice, lcupar, cupd**; radiative processes **lrad, dtrad, njump, lcool**; vertical advection of TKE, rain and snow **dxmeso, dzqlim**

4.2.2.3. Numerical Damping

To control numerical instability and the growth of unrealistic tendencies, several damping features have been added to the COAMPS code.

Input parameters:

diffusion and filtering **ldiff, dif2nd, dif4th, lfft**; divergence damping **lddamp, divdmp**; time filtering **robert**; upper level sponge layer **lralee, rdtime, nrdamp, lspong**

4.2.2.4. Additional Model Features

These parameters provide options for the advective scheme (e.g. 2^{nd} order, 4^{th} order, flux or advective schemes) and allow for adding a large scale pressure gradient via the geostrophic wind.

Input parameters:

advection scheme **iadvct**; geostrophic wind **lvgeo, ugeo, vgeo**

4.2.3. Data Manipulation

Several namelist parameters are used to control the flow of data into and out of the COAMPS system. These inputs are categorized into three groups: data input and output parameters (used for analysis, restarts and data assimilation purposes), graphics parameters (used to view model output), and diagnostic parameters (used to numerically evaluate model results).

Input parameters:

directory locations **dsngff, dsnrff**; times for printing/saving output **kprnta, ksavea, ksaves, isavefrq, ifsave**; data format type **lwritu**; 1D graphics output **dset1d, t1d, i1d(nn), j1d(nn), cpar(nn), city(nn)**; 2D graphics output **npfil, xsfil, lwritu**; 3D graphics

output **tvis1, tvis2, vpar1, vpar2, lprvis, prsvis, hgtvis, ksaves, lwritu**; diagnostic output **laper, lasym, ldgadj, lcuppr, lprnti**

Although most of these output times are designated by time intervals, note that parameter **ksaves** is an array containing actual times during the forecast that 3D volumes of sigma level data are to be saved. Caution is advised when outputting 3D volumes over a long forecast period or at many times, especially for more than one mesh since they are quite large (a (65x65x30) grid requires approximately 100Mb of space).

Having reviewed the more important COAMPS input parameters in this section, and using Appendix B as an example of a COAMPS input file, the user is now able to create a COAMPS input file that sets up any type of model run. The final step in performing a COAMPS simulation is to submit the run job as described in the next section.

4.3. Submitting a COAMPS Job

The user is ready to submit a COAMPS job once the following steps have been completed:

1. The analysis and forecast codes have been compiled and linked to form the COAMPS executables (*coamps_analysis.exe* and *coamps_forecast.exe*.
2. The data has been obtained to specify the initial input fields.
3. The COAMPS run script *run.coamps* has been edited to update the namelist input parameters (i.e. the namelist input file *nlist.input* has been modified to suit the particular process study).

In the example given below, the COAMPS run script template, *run.coamps*, has been modified and renamed *run.runsd* where *runsd* is a user-created subdirectory under */run*. This script creates a COAMPS namelist input file called *nlist.$ddtg.runsd* where *$ddtg* is the a 10 digit date-time group (YYYYMMDDHH) provided to the script file as an input argument.

Within the */run/runsd* subdirectory, submit the run script by typing: "**run.runsd $ddtg 3**". The first input argument represents the starting date-time group. The second is a run action argument indicating whether the analysis or forecast is to be run.

Table 6: Options for the 2nd run script input argument

ARGUMENT	ACTION
0	no action
1	run analysis
2	run forecast
3	run analysis and forecast

When the second input argument is '1' or '3', the run script executes the analysis program using the following command:

$HOME/coamps/bin/coamps_analysis.exe < nlist.$ddtg.runsd > output.a.$ddtg

The user input is redirected into the analysis program from file *nlist.$ddtg.runsd* and standard output/error is redirected out to file *output.a.$ddtg*. Upon completion, the directory specified by namelist input **dsnrff** contains the analysis fields produced by the executable. Scan this directory and view the file *output.a.$ddtg* to ensure that the analysis completed successfully.

Similarly, when the second input argument is '2' or '3', the run script executes the forecast using the following command:

$HOME/coamps/bin/coamps_forecast.exe < nlist.$ddtg.runsd> output.m.$ddtg

User input is redirected into the forecast program from file *nlist.$ddtg.runsd* and standard output/error is redirected out to file *output.m.$ddtg*. Upon successful completion of the COAMPS forecast, user-requested forecast fields reside in the directories specified as namelist input **dsnrff** and **dset1d**. These data may be viewed graphically with a variety of display packages that process 1D, 2D or 3D model output the details of which are covered in a separate document. For a quick analysis of the COAMPS results at time intervals given by the user input parameter **kprnta**, the user may look at 'quick prints' of horizontal and vertical slice numerical model output embedded in the standard output/error file *output.m.$ddtg*. The output to this file also indicates whether the job has completed successfully.

A data assimilation update cycle may be submitted by inputting two additional run script arguments. The third input argument is a 10 digit ending date-time group (YYYYMMDDHH) and the forth argument is the number of hours between updates (typically 12 hours). To submit a 12 hour data assimilation cycle, the user types: "**run.runsd $ddtg 3 $enddtg 12**". The command initiates a forecast starting at **$ddtg** and continuing through **$enddtg** assimilating data every 12 hours. This type of run is only appropriate when observations are available every 12 hours throughout the time period since these data are needed to adjust the first guess fields.

Appendix A: Sample makefile

This *Makefile* template is from the */coamps#/mod/Templates* directory that is created by running the script file *get.templates* for the SGI machine Rossby.

file *Makefile*

```
#######################################################################
#######################################################################
#   makefile to build program coamps_analysis.exe and coamps_forecast.exe
#######################################################################
#######################################################################

#######################################################################
#   other dependencies
#######################################################################

include /data_rossby6/users/coamps/coamps2.0/config.os

include /data_rossby6/users/coamps/coamps2.0/config.user

LIBDIR   = /data_rossby6/users/coamps/lib/2.0
MAINANL  = /data_rossby6/users/coamps/coamps2.0/src/coamps_analysis/coamps_analysis.F
MAINFCST = /data_rossby6/users/coamps/coamps2.0/src/coamps_forecast/coamps_forecast.F
INCL_PROL_MAIN = /data_rossby6/users/coamps/coamps2.0/prologues/main
INCL_PROL_COAMPS = /data_rossby6/users/coamps/coamps2.0/prologues/coampslib
INCL_PROL_OI = /data_rossby6/users/coamps/coamps2.0/prologues/oilib

LIBS    = $(LIBDIR)/coampslib.a \
          $(LIBDIR)/fnoclib.a   \
          $(LIBDIR)/oilib.a     \
          $(LIBDIR)/coampslib.a \
          $(LIBDIR)/nl_beqlib.a \
          $(LIBDIR)/fishpaklib.a

MAKEFILE = Makefile

# OBJS: include the coamps .o files that the user modified

OBJS =

PROGRAMS = coamps_analysis.exe \
           coamps_forecast.exe

target : $(PROGRAMS) $(OBJS) $(LIBS)                                    (continued)
```

```
#####################################################################
#####################################################################
#  loader directive
#####################################################################
#####################################################################

coamps_analysis.exe : $(MAINANL) $(OBJS) $(LIBS)
        $(RM) $@
        $(LD) $(LDFLAGS) -o $@ -I$(INCL_PROL_MAIN) $(MAINANL) $(LIBS) $(ISISLIBS)
$(EXTRALIBS)
        mv $@ ../../bin/$@

coamps_forecast.exe : $(MAINFCST) $(OBJS) $(LIBS)
        $(RM) $@
        $(LD) $(LDFLAGS) -o $@ -I$(INCL_PROL_MAIN) $(MAINFCST) $(LIBS) $(ISISLIBS)
$(EXTRALIBS)
        mv $@ ../../bin/$@

.F.o:
        $(RM) $@
        $(FC) -c $(FFLAGSM) -I$(INCL_PROL_COAMPS) -I$(INCL_PROL_OI) $*.F

#####################################################################

#####################################################################
# common rules for all Makefiles - do not edit
#####################################################################

emptyrule:

clean:
        $(RM_CMD) "#"*

#####################################################################
#  done
#####################################################################
```

Appendix B: Sample COAMPS user input file *nlist.input*

file *nlist.input*

```
&gridnl

     kka   = 30,
     ma    = 65, 73, 109,
     na    = 65,  73,  91,
     nbdypt =  7,
     nnest  =  3 ,

    &end

&coamnl

     dsclim = 'insert directory path of climatology data',
     dsdted = 'insert directory path of 1km terrain data',
     dsgiss = 'insert directory path of Goddard Institute of Space Studies data',
     dsngff = 'insert directory path of NOGAPS flat files',
     dsnrff = 'insert directory path of COAMPS flat files',

     npfil  = 'insert name of ocards file: 2D horizontal graphics directives' ,
     xsfil  = 'insert name of xcards file: 2D vertical graphics directives',

     icase =    0,
     ibdya =    7,
     jbdya =    7,
     kgetbc =  12,
     nproj =    2,
     alnnt =    5.0,
     phnt1 =   60.0,
     phnt2 =   30.0,
     rlat1 =   45.0,
     rlong1 =   5.0,
     iref   =  33,
     jref   =  33,
     ii     =  21, 20
     jj     =  18, 17
     delx =  81000.0,
     dely =  81000.0,
     delta =   240.0,
     ktaust =  0, 0, 0,
     ktauf     =  24, 0, 0,
                  24, 0, 0,
                  24, 0, 0,
     mtaua =   2,
     alpha =    1.0,
     dsigma =  7500.0, 5800.0, 4200.0, 2500.0, 1000.0,
               1000.0,  750.0,  750.0,  750.0,  750.0,
               750.0,  750.0, 1000.0, 1000.0, 1000.0,
               1000.0,  800.0,  800.0,  800.0,  600.0,
               400.0,  300.0,  200.0,  140.0,   90.0,
               60.0,   40.0,   30.0,   20.0,   20.0,
```

(continued)

```
ldiff  = t,
dif2nd =     0.04,
dif4th =    -0.005,
lddamp = t,
divdmp =    50.0,
lralee = f,
rdtime =   150.0,
nrdamp =   8,
robert =    0.2,
l2way = f,
lvgeo  = f,
iadvct =   2,
cbar   =   240.0,

itauin =  12,
itauf  =  24,
icycle =  12,
iupd   =  0,
loi    = f,
loimf  = f,
lnrhty = f,
lqanl = f,
ltanl = f,
lpseud = f,
ivar = 0

lcupar = t,
lcuppr = f,
lmoist = t,
lice   = t ,
lrad   = t,
njump = 1,
lcool  = f,
lsfcen = t,
lflux  = t,
linit  = t,

lles   = f,
ltke   = t ,
almin  =    0.02,
al0    =    1.0,
sh0    =    0.675,
sm0    =    0.5,
iahsgm =  0,
iamxgl =  4,
iashsm =  4,

ilndflg = 1,
ialbflg = 1,
iz0flg = 1,
igwtflg = 1,
iseaflg = 1,
itopoflg = 1,
ltopoa = t,
```

(continued)

```
    nftopo =   3,

    kprnta =  48, 0, 0,
    ksavea =  144, 0, 0,
    ifsave = 2
    ksaves =   24, 0,  0,
    lwritu = t,
    lprnti = f,
    locard = f,
    ldbms  = f,
    ldbmsi = f,
    ldbmso = f,

   &end

&mvoinl

    datfil = 'insert directory path of observational data'

   &end
```

Appendix C: Input parameters: *gridnl* namelist

VAR	DEFAULT VALUE	DEFINITION
INTEGER VARIABLES		
kka	30	Number of vertical height levels in the atmosphere **NOTE:** (**kka** must be equal to the number of **dsigma** values setup in the namelist)
lm	16	Number of vertical pressure levels on which real data initial conditions are specified
ma	61	1D array: Number of grid points in the x direction for each mesh **NOTE:** (for inner nests (**ma**-1)/3 must be a whole number)
na	61	1D array: Number of grid points in the y direction for each mesh **NOTE:** (for inner nests (**ma**-1)/3 must be a whole number)
nbdya	7	Real data lateral boundary condition identifier for inner nests: (6) Perkey-Krietzberg (7) Davies
nbdypt	5	Number of points included in lateral boundary computation **NOTE:** (**ma** and **na** > (2***nbdypt**))
nnest	1	Total number of meshes

Appendix D: Input parameters: *coamnl* namelist

VAR	DEFAULT VALUE	DEFINITION
CHARACTER VARIABLES		
city	' '	Character*32, 2D array: City names associated with grid point locations (**i1d,j1d**) for each mesh; used for labeling 1D graphics output (*get1d.F, asav1d.F*)
cpar	' '	Character*4, 2D array: 1D graphics variable identification string for each mesh; used to select variables for 1D graphics output (*get1d.F, asav1d.F*)
dsclim	'/home/hodur/masterr/'	Character*80: Directory path of climatology data; used to initialize the real data surface parameters (**albedo, sst, grdwet, rland, zsfc, z0**) when high resolution data not available (*sfcpar.F, soiltp.F*)
dsdted	'/u/c/hodur/masdted/'	Character*96: Directory path of 1 km terrain data; used for horizontal resolution < 20 km and if available (*sfcpar.F*)
dsgiss	'/a/ops/etc/static/met /noraps/masgiss/'	Character*80: Directory path of Goddard Institute Space Studies high resolution **albedo** and **z0** data (*sfcpar.F*)
dsetng	' '	Character*24: ISIS directory path of NOGAPS data input/output; used for location of boundary conditions and/or cold start initial conditions when **ldbms**=.true. (*gethty.F, getng.F, getsst.F, inhesc.F, inngsc.F, innguv.F*)
dsetg	' '	Character*24, 1D array: ISIS directory path of COAMPS data input/output for each mesh; used for location of COAMPS files when **ldbms**=.true. (*gethty.F, getng.F, getsst.F, inhesc.F, inngsc.F, innguv.F*)
dset1d	'./'	Character*80: Directory path for 1D graphics output (*get1d.F, asav1d.F*)
dsngff	' '	Character*80: Directory path of NOGAPS flat files; used for location of boundary conditions and/or cold start initial conditions when **ldbms**=.false. (*gethty.F, getng.F, getsst.F, inngsc.F, innguv.F*)
dsnrff	' '	Character*80: Directory path of COAMPS flat files; used for location of COAMPS input/output needed for incremental update, model restart, and 2D and 3D graphics output when **ldbms**=.false. (*coamm.F, io*.F*)
gnamg	' '	Character*32, 1D array: ISIS geometry name for each mesh; used by ISIS when **ldbms** and **ldbmsi**=.true.
mdltg	' '	Character*32, 1D array: ISIS model type for each mesh; used by ISIS when **ldbms** and **ldbmsi**=.true.
npfil	'ocards'	Character*80: name of file containing 2D horizontal

VAR	DEFAULT VALUE	DEFINITION
		level plotting information (fields, plot type, level, time, mesh number, etc.)
xsfil	'xcards'	Character*80: name of file containing 2D cross-sectional plotting information (fields, plot type, levels, time, mesh number, etc.)
vpar1	' '	Character*4, 2D array: 3D graphics variable identification string set 1 for each mesh; used to select *Visual* output (*visout.F*)
vpar2	' '	Character*4, 2D arrays: 3D graphics variable identification string set 2 for each mesh; used to select *Visual* output (*visout.F*)
REAL VARIABLES		
albdo	0. 2	Idealized value of surface refractivity; used to specify the albedo when **ialbflg** = 0 (*user_sfce.F*) **NOTE:** (0.0 = low refractivity, 1.0 = high refractivity)
alndpct	1.0	Idealized land/water indicator; used to specify whether a point is a land or water point when **ilndflg** = 0 (*user_sfc.F*) **NOTE**: (0.0 = water, 1.0 = land)
almin	0.02 m	Minimum mixing length factor (*afore.F*)
alnnt	240.0 °	Longitude aligned north-south; used for rotating a lambert conformal/polar stereographic projection (*grid.F*) **NOTES:** (increases eastward from 0° to 360° beginning at Greenwich, England) (longitudes outside the model domain may be used to rotate the grid)
alpha	0.4	Averaging coefficient for semi-implicit time differencing (*alhs.F, alhsin.F*).
al0	100.0 m	Mixing length constant/limiting value (*coamm.F*)
cbar	100.0 m/s	Speed of sound **cbar**=$\sqrt{c_p R\overline{T_v} / c_v}$ (*alhs.F, alhsin.F, asetim,f*) **NOTES:** (if **cbar**≤0, value computed in *coamm.F*) (because of numerical techniques, **cbar** ≥ 2*|V|)
climrh	.01,.01,.01,.02,.02,.10,.15 ,.20,.25,.30,.40,.50,.60,.70 ,.70,.70,84*.70	1D array: Climatological values of relative humidity at user specified pressure (**pr**) levels (*getng.F*).
cstar	300.0 m/s	Speed of gravity waves **cstar** = $(z_T/\pi)\sqrt{(g/\theta_v)\partial\theta_v/\partial z}$ where z_t the is domain top; used for radiative lateral boundaries (**ibdya=jbdya**= 4 or 5) (*bwind.F*)
cupdx	10000.0 m	Minimum horizontal resolution for Kain-Fritsch convective parameterization scheme (*amodel.F*)
delta	2.0 s	Large, coarse mesh time step; used for integration of slow modes **NOTE:** (cfl condition: **delta** ≤ **dsigma**/w & **delta** ≤ 0.7***delx**/(2*|V|)) (inner nest time steps

VAR	DEFAULT VALUE	DEFINITION
		computed internally)
delx	40000.0 m (specify in degrees of longitude when nproj=5)	Coarse mesh grid spacing in the x-direction; used for computing horizontal spatial derivatives **NOTE:** (inner nest spacing computed internally)
dely	40000.0 m (specify in degrees of latitude when nproj=5)	Coarse mesh grid spacing in the y-direction; used for computing horizontal spatial derivatives **NOTE:** (inner nest spacing computed internally)
dif2nd	0.01	2nd order dimensionless diffusion coefficient **dif2nd**=$K_{d2}\Delta t/\Delta x^2 \leq 1/8$ for stability; used to control small scale numerical noise (*asetup.F*)
dif4th	-0.0025	4th order dimensionless diffusion coefficient **dif4th**=$K_{d4}\Delta t/\Delta x^4 \geq -1/32$ for stability; used to control small scale numerical noise (*asetup.F*) **NOTE:** (if non- negative, 2nd order diffusion is used)
divdmp	10.0	Dimensionless divergence damping coefficient **divdmp**=$K_{div}\Delta\tau/\Delta x^2 \leq 1/2$ for stability; used to control amplification of the acoustic mode (*alhs.F*) **NOTE:** (smaller values produce greater damping)
dsigma	500.0 m	1D array: Vertical grid spacing in the atmosphere; used to compute vertical spatial derivatives **NOTE:** (non uniform vertical spacing is permitted)
dtrad	3600.0 s	Radiation time interval; used to determine frequency of radiation computations (*coamm.F*)
dxmeso	9000.0 m	Upper limit of horizontal resolution below which the vertical advection of tke, raindrops and snow are included in prognostic equations (*afore.F, aforqx.F*)
dzqlim	500.0,250.0,100.0,100.0, 100.0,100.,100 m	Upper limit of height, for each mesh, below which vertical spacing for vertical advection of rain and snow is computed (*dzqrs.F*)
flat	0.0 °	Latitude; Used for Cartesian grids (**nproj**=4) to compute the Coriolis parameter (*grid.F*)
gwetmx	1.0	Maximum ground wetness (sfcpar.f)
hgtsnd	999.0 m	1D array: Heights above surface of idealized sounding data (*refsnd.F*) **NOTE**: (computed internally)
hgtvis	10., 50., 100., 200., 300., 400., 500., 750., 1000.,1250., 1500., 2000., 2500.,3000., 3500., 4000., 4500.,5000., 6000., 7000., 8000.,9000.,10000.,12000.,15000.,75*0.0 m	1D array: Heights above the surface that 3D graphics data are interpolated to; used for *Visual* output when **lprvis**=.false. (*visout.F*)
phnt1	30.0°	Latitude where map projection intersects the earth and map factor is 1.0; used to setup map projection (*grid.F*)

VAR	DEFAULT VALUE	DEFINITION
		NOTE: (varies from -90°at the south pole to +90° at the north pole)
phnt2	60.0°	2nd Latitude where map projection intersects the earth; used to setup lambert conformal map projection(*grid.F*) **NOTE:** (varies from -90°at the south pole to +90° at the north pole)
pr	10.0, 20.0, 30.0, 50.0 , 70.0, 100.0,150.0, 200.0 ,250.0, 300.0,400.0, 500.0 ,700.0, 850.0,925.0,1000.0 , 84*0.0 mb	1D array: Pressure levels corresponding to analyzed fields: used to obtain and generate real data initial conditions (*getng.F, getfg1.F, inlvl.F*)
prk	0.0	Computed internally
prsvis	1030.,1000.,975.,950.,925. , 900., 875.,850.,825.,800. , 775., 750.,725.,700.,650. , 600., 550.,500.,400.,350. , 300., 250.,200.,100. , 76*0.0 mb	1D array: Pressure levels that 3D graphics data are interpolated to; used for *Visual* output when **lprvis**=.true. (*visout.F*)
psfc0	100000.0 kg/ms	Surface pressure; used to compute terrain pressure in idealized cases (*user_sfc.F*)
psnd	-999.0	1D array: Pressure or height levels of idealized sounding data (*refsnd.F*) **NOTE:** (pressure in mb when **ipsnd**=1, height in meters when **ipsnd**=2)
qsnd	-999.0	1D array: Moisture of idealized sounding data (*refsnd.F*) **NOTE:** (moisture in form of dewpoint temperature degrees Celcius when **iqsnd**=1, dewpoint temperature degrees Kelvin when **iqsnd**=2, mixing ratio (g/kg) when **iqsnd**=3, relative humidity (%) when **iqsnd**=4)
rdtime	120.0 s	Time scale for damping u,v,w,θ in sponge layer; used to control vertical reflections (*coamm.F*) **NOTES:** (the larger the time scale, the weaker the damping) (**rdtime** $\approx 1/(3.5\,w\,(2\pi/L_z))$ where L_z is the vertical wavelength)
rlat1	40.0°	Reference latitude, corresponds to reference point (**iref(1),jref(1)**); used to setup the coarse grid (*grid.F*) **NOTE:** (varies from -90°at the south pole to +90° at the north pole)
rlong1	240.0°	Reference longitude, corresponds to reference point (**iref(1),jref(1)**); used to setup the coarse grid (*grid.F*) **NOTE:** (varies from -90°at the south pole to +90° at the north pole)

VAR	DEFAULT VALUE	DEFINITION
robert	0.05	Coefficient for coupling the three time levels; used to filter high frequency oscillations (*amodel.F, atimef.F*) **NOTE:** (suggested value of **robert**=0.05 when **lddamp**=.true., otherwise, **robert**=0.1-0.2)
seatmp	285.0 K	Idealized surface temperature; used to specify the surface temperature when **iseaflg** = 0 (*user_sfc.F*)
sdmult	0.0	Multiplication factor for standard deviation of terrain height (*sfcpar.F*)
sfcwet	0.1	Idealized value for ground wetness; used to specify the ground wetness when **igwtflg** = 0 (*user_sfc.F*) **NOTE:** (0.0 = saturated, 1.0 = dry)
sh0	0.675	TKE constant; used to compute the TKE coefficient **sh** when **iashsm**=2,3,4 (*afore.F*) **NOTE:** (typically **sh0**=0.675 when $\Delta x >> \Delta z$; **sh0**=0.3 when $\Delta x \cong \Delta z$)
sm0	0.5	TKE constant; used to compute the TKE coefficient **sm** when **iashsm**=2,3,4 (*afore.F*) **NOTE:** (typically **sm0**=0.5 when $\Delta x >> \Delta z$; **sm0**=0.1 when $\Delta x \cong \Delta z$)
strgrd	1	Factor for stretching mesh horizontally (*grid.F*)
tamp	1.5	Amplitude of idealized temperature perturbation (*flds08.F*)
t1d	3600.0 s	1D graphics time interval; used to determine frequency of *coam1d* output (*asav1d.F*)
tvis1	0.0 s	1D array: 3D graphics time interval for each mesh; used to determine frequency of *Visual* set 1 output (*visout.F*)
tvis2	0.0 s	1D array: 3D graphics time interval for each mesh; used to determine frequency of *Visual* set 2 output (*visout.F*)
tsnd	-999.0	1D array: Temperatures of idealized sounding data (*refsnd.F*) **NOTE:** (temperature in degrees Kelvin when **itsnd**=1, in degrees Celcius when **itsnd**=2, in theta when **itsnd**=3)
ugeoa **vgeoa**	0.0 m/s	1D arrays: Geostrophic wind components; used to compute the synoptic scale pressure gradient force when **lvgeo**=.true. (*arhsu.F, arhsv.F*)
umean **vmean**	0.0 m/s	Constant values representing a mean wind; used to subtract a mean wind from the idealized sounding data (*istate.F*)
usnd **vsnd**	-999.0 m/s	1D arrays: Winds of idealized sounding data (*refsnd.F*) **NOTE:** (winds in terms of components (m/s) when **iuvsnd**=2, in terms of direction-**usnd** and speed-**vsnd** when **iuvsnd**=1)
z0z0mx	0.1 m	Maximum surface roughness (*sfcpar.F*)

VAR	DEFAULT VALUE	DEFINITION
zrough	0.01 m	Idealized value for surface roughness; used to specify the surface roughness when **iz0flg** = 0 (*user_sfc.F*) **NOTE:** (0.0 = smooth, 1.0 = rough)
INTEGER VARIABLES		
iadvct	1	Advection type identifier (*arhs*.F, afor*.F, advsc.F*): (1) 4th order advective (2) 2nd order advective (3) 2nd order advective with 2nd order upstream for θ (4) spectral (5) 2nd order flux (6) mixed: 2nd order advective and flux
ialbflg	1	Indicator for use of idealized albedo data for each mesh (*user_sfc.F*) (**NOTE:** albedo given by idealized value (**albdo**) when **ialbflg**=0, otherwise uses database)
iahsgm	1	Horizontal subgrid scale mixing identifier: (*arhs*.F, afor*.F, hmix*.F*) (1) eddy mixing coefficient based upon deformation field (use when $\Delta x >> \Delta z$) (2) eddy mixing coefficient based upon TKE prediction (use when $\Delta x \cong \Delta z$)
iamxgl	1	Mixing length identifier: options for the mixing lengths **al1, al2, ald** are given in Table D-1 below (*afore.F,coamm.F*)
iashsm	1	TKE coefficients **sm** and **sh** identifier; used to define eddy mixing coefficients (*afore.F*): (1) Ri dependent (Mellor & Yamada, 74) (2) **sm=sm0, sh=sh0** (3) **sm=sm0, sh=**(1.0+2.0 **al2/al1**)***sm0** (4) **sm=sm0, sh=**2.13***sm0** (5) Ri dependent (Mellor & Yamada, 82)
ibdya	1	x-direction boundary condition identifier (*abdy*.F*): **idealized** (1) fixed (2) extrapolated (3) periodic (4) radiation (5) radiation and extrapolation **real data** (6) Perkey-Krietzberg (7) Davies
icase	1	Case number identifier; used to indicate real or idealized data (*coama.F, flds##.F*): (0) real data (##) value corresponds to the number in *flds##.F*

VAR	DEFAULT VALUE	DEFINITION
		filename, a file where perturbations of the mean state are specified
icycle	0 hr.	Frequency of a data assimilation cycle; used to read/write input/output fields (*coama.F, coamm.F*)
idgadj	2	x-grid point; used for printout of microphysical diagnostics (*adjtq.F*)
ifsave	1	Indicator for saving sigma level data (*coamm.F*): (1) at specific times given by **ksaves** (2) at time intervals given by **isavfrq**
igwtflg	1	Indicator for use of idealized ground wetness for each mesh (*user_sfc.F*) (**NOTE:** ground wetness given by idealized value (**sfcwet**) when **igwtflg**=0, otherwise uses database)
ii	21	1D array: Parent meshes x-grid point corresponding to left boundary of next inner nest; used to setup inner nests (*coama.F*)
ilndflg	1	Indicator for use of idealized land/sea grid point for each mesh (*user_sfc.F*) (**NOTE:** land/sea point given by idealized value (**alndpct**) when **ilndflg**=0, otherwise uses database)
iloi	0	MVOI volume type identifier (*coama.F*): (0) ivol, jvol = number of MVOI volumes (1) ivol, jvol = distance in kilometers of an MVOI volume
ilsp	1	Indicator for use of explicit moist physics for each mesh (*amodel.F*) **NOTE:** (explicit moist physics when **ilsp**=1, large scale precip param when **ilsp**=2)
ipsnd	1	Indicator of units of idealized pressure data (*refsnd.F*) **NOTE:** (pressure in mb when **ipsnd**=1, height in meters when **ipsnd**=2)
iqsnd	1	Indicator of units of idealized moisture data (*refsnd.F*) **NOTE:** (moisture in form of dewpoint temperature degrees Celcius when **iqsnd**=1, dewpoint temperature degrees Kelvin when **iqsnd**=2, mixing ratio (g/kg) when **iqsnd**=3, relative humidity (%) when **iqsnd**=4)
iref	55	1D array: x-grid point on each mesh at reference latitude **rlat1**; used to setup the model domains (*grid.F*) **NOTE:** (**iref(1)** can be larger than **ma(1)** when **alnnt** and **rlong1** fall outside the model domain -- for rotated grids) (**iref** for inner nests computed internally)
isavefrq	-1,0,0 hr,min,sec	Times for saving sigma level data; used to graphically view COAMPS forecast fields (*coamm.F*) **NOTE:**

VAR	DEFAULT VALUE	DEFINITION
		(used when **ifsave**=2)
iseaflg	1	Indicator for use of idealized surface temperature data for each mesh (*user_sfc.F*) (**NOTE:** surface temperature given by idealized value (**seatmp**) when **iseaflg**=0, otherwise uses database)
istr1	1	Factor for horizontally stretched grid (*grid.F*)
istr2	1	Factor for horizontally stretched grid (*grid.F*)
itopoflg	1	Indicator for use of idealized terrain data for each mesh (*user_sfc.F*) (**NOTE:** topography height set to zero when **itopoflg**=0, otherwise uses database)
itauf	12 hr.	Ending time for obtaining and writing out coarse mesh boundary tendencies; used for data assimilation (*coama.F*)
itauin	12 hr.	Initial time (and frequency) for obtaining and writing out coarse mesh boundary tendencies from NOGAPS fields; used for data assimilation (*coama.F*) **NOTE:** (value should be equivalent to forecast interval of NOGAPS fields)
itsnd	1	Indicator of units of idealized temperature data (*refsnd.F*) **NOTE:** (temperature in degrees Kelvin when **itsnd**=1, in degrees Celcius when **itsnd**=2, in theta when **itsnd**=3)
iupd	0	Data assimilation identifier (*coama.F, asetup.F, inlvl.F, coamm.F*): (0) no update; analyzed NOGAPS fields on p levels (1) full update; analyzed COAMPS fields on p levels (2) incremental update; COAMPS fields on sigma-z levels with MVOI increments interpolated to sigma-z levels
iuvsnd	1	Indicator of units of idealized wind data (*refsnd.F*) **NOTE:** (winds in terms of components (m/s) when **iuvsnd**=2, in terms of direction-**usnd** and speed-**vsnd** when **iuvsnd**=1)
ivar	0	Nonlinear variational balancing identifier; used in MVOI analysis (*oi_anl.F*): (0) no balancing (1) balancing
ivol	5	1D array: Number of regions (MVOI volumes) that each mesh is divided into in the x-direction; used for MVOI analysis (*oivol.F, oi_anl.F*)
i1d	0	2D array: 1D graphics x-grid points for each mesh; used to specify location of 1D graphics output (*asav1d.F*)

VAR	DEFAULT VALUE	DEFINITION
iz0flg	1	Indicator for use of idealized surface roughness data for each mesh (*user_sfc.F*) (**NOTE:** surface roughness set to idealized value (**zrough**) when **iz0flg**=0, otherwise uses database)
jbdya	1	y-direction boundary condition identifier (*abdy*.F*): **idealized** (1) fixed (2) extrapolated (3) periodic (4) radiation (5) radiation and extrapolation **real data** (6) Perkey-Krietzberg (7) Davies **NOTE:** (in general **ibdya=jbdya**)
jdgadj	2	y-grid point: used for printout of microphysical diagnostics (*adjtq.F*)
jref	42	1D array: y-grid point at reference longitude **rlong1**; used to setup the model domains (*grid.F*) **NOTE:** (**jref** for inner nests computed internally)
jj	21	1D array: Parent meshes y-grid point corresponding to lower boundary of next inner nest; used to setup inner nests (*coama.F*)
jqq	550	Max number of observations per MVOI volume (*oi_anl.F*)
jvol	5	1D array: Number of regions (MVOI volumes) that each mesh is divided into in the y-direction; used for MVOI analysis (*oivol.F, oi_anl.F*)
j1d	0	2D array: 1D graphics y-grid points for each mesh; used to specify location of 1D graphics output (*asav1d.F*)
kgetbc	12 hr.	Frequency for updating coarse mesh boundary tendencies; used in data assimilation (*coamm.F*) **NOTE:** (value chosen so that **kgetbc/itauin** is a whole number)
kprnta	6,0,0 hr,min,sec	Frequency for writing forecast fields to standard output; used for viewing slices of numerical model data (*coamm.F*)
ksavea	-1,0,0 hr,min,sec	Frequency for writing restart files to directory **dsnrff**; used to obtain fields used for model restart at a nonzero time (*coamm.F*)
ksaves	-1 hr,min,sec	2D array: Times for saving sigma level data for each mesh; used to graphically view COAMPS forecast fields (*coamm.F*) **NOTE:** (used when **ifsave**=1)

VAR	DEFAULT VALUE	DEFINITION
ktauf	12,0,0 hr,min,sec	2D array: Ending time of forecast for each mesh (*coamm.F*)
ktaust	0,0,0 hr,min,sec	Beginning time of forecast (coamm.f) **NOTE:** (**ktaust** usually equal to zero except for a model restart – see **ksavea** to save fields for a model restart)
mtaua	5	Number of small coarse mesh time steps per (**delta**) large time steps; used for integration of fast modes (*alhs.F*) **NOTE:** (cfl condition: **deltaua** ≤ **dsigma**/w & **deltaua** ≤ 0.7***delx**/(**cbar**+|V|)) (inner nest time steps computed internally) (because of numerical techniques used, **mtaua** typically set to 2)
nftopo	3	Number of passes through 25-point filter; used to smooth terrain field (*sfcpar.F*)
nfpass	3	Number of passes through 9- or 25-point filter; used to smooth output and analysis fields (*filt9.F , filt25.F*)
njump	3	**njump**-1 is number of x- and y-grid points skipped in radiation computations (*radiat.F*)
noisub	12	Number of processors; for MVOI multitasking (*oi_anl.F*)
nproj	2	Map projection identifier; used to setup the horizontal domain (*grid.F*): (1) mercator (2) lambert conformal (3) polar stereographic (4) Cartesian coordinates (5) spherical
nrdamp	8	Number of upper model levels that are in the sponge layer; used to damp vertical reflections (*coamm.F*) **NOTE:** (8 or more layers recommended)
LOGICAL VARIABLES		
laper	.false.	Check for aperiodicity at boundaries; used as a check for idealized cases with periodic boundary conditions (**ibdya=jbdya**=3) (*asym1.F, asym2.F*): **true** - check periodicity **false** - do not do above
lasym	.false.	Check for asymmetry in fields; used as a check for idealized cases with symmetric initial conditions and forcing (*amodel.F*): **true** - check for asymmetry **false** - do not do above
lcool	.false.	Simple radiational cooling (*aforth.F*): **true** - cool 2° per day below 10 km **false** - do not do above

VAR	DEFAULT VALUE	DEFINITION
lcupar	.false.	Kain-Fritsch convective parameterization (*amodel.F*): **true** - perform convective adjustments to (θ, q_v, q_c, q_r) on meshes with resolution Δx>**cupdx** **false** - do not do above
lcuppr	.false.	Kain-Fritsch parameterization diagnostics (*kfpara.F*): **true** - print diagnostics **false** - do not do above
ldbms	.false.	Read NOGAPS initial conditions (*coama.F*): **true** - ISIS database **false** - flat files
ldbmsi	.false.	Read/write COAMPS input/output (*coama.F, coamm.F*): **true** - ISIS database **false** - flat files
ldbmso	.false.	Write COAMPS analysis fields (*coama.F*): **true** - ISIS database **false** - flat files
lddamp	.false.	Divergence damping; used to reduce accumulation of mass within a grid volume (*alhs.F*): **true** - use divergence damping **false** - do not do above
ldgadj	.false.	Microphysical diagnostics (*adjtq.F*): **true** - print diagnostics **false** - do not do above
ldiff	.true.	Numerical diffusion (*arhs*.F, afor*.F*): **true** - use diffusion **false** - do not do above
lfftf	.false.	FFT filter -- spectral truncation of wavelengths $\leq 3\Delta x$; used in place of numerical diffusion for LES cases with periodic boundary conditions (**ibdya=jbdya**=3) (*fftfil.F*): **true** - use FFT filter (requires that **ldiff** =.false. also) **false** - do not do above
lflux	.true.	Surface fluxes (*sfcflx.F*): **true** - nonzero surface heat, moisture, momentum flux **false** - zero surface heat, moisture, momentum flux
lles	.false.	Large eddy simulation; use when $\Delta x \equiv \Delta z$ (*afore.F, sfcflx.F*): **true** - use terms and coefficients in the TKE and surface flux computations that are appropriate for 3D turbulence **false** - do not do above
lice	.true.	Ice physics (*adjtq.F, amodel.F*):

VAR	DEFAULT VALUE	DEFINITION
		true - temporal variations and adjustments computed for frozen moisture variables (q_i, q_s) (requires that **lmoist**=.true. also) **false** - set frozen moisture variables to zero
linit	.false.	Initialize certain variables (*ainit.F*): **Real data** **true** - eliminate net inflow/outflow, compute vertical velocity **false** - do not do above **Idealized data** (based upon value of icase) **true** - initialize pressure, or potential temperature, or add perturbations to horizontal winds **false** - do not do above
lmoist	.true.	Moist physics (*adjtq.F, amodel.F*): **true** - temporal variations and adjustments computed for liquid moisture variables (q_c, q_r) and frozen moisture variables (q_i, q_s) when **lice**=.true. also) **false** - set liquid and frozen moisture variables to zero
lnrhty	.true.	Initialization of surface fields (*gethty.F, nethty.F*): **true** - use COAMPS history fields if available **false** - use other source: NOGAPS, climatology fields
locard	.false.	Read in 2D horizontal level plotting information (*coama.F, coamm.F*): **true** - ISIS operational format **false** - do not do above
loi	.false.	Multivariate optimum interpolation (MVOI) analysis (*oi_anl.F, anlfld.F*): **true** - perform MVOI **false** - do not do above
loimf	.false.	Inner nest MVOI analysis (*oi_anl.F, anlfld.F*): **true** - perform MVOI on inner nest fields **false** - interpolate coarse mesh increments to inner nest fields
lprint	.false.	Surface parameter diagnostics (*sfcpar.F*): **true** - print diagnostics **false** - do not do above
lprvis	.false.	*Visual* data vertical interpolation (*visout.F*): **true** - pressure levels given by the **prsvis** array **false** - height levels (above surface) given by the **hgtvis** array
lqanl	.false.	Moisture analysis (*anlfld.F, oi_anl.F*): **true** - perform analysis of initial moisture field **false** - do not do above

VAR	DEFAULT VALUE	DEFINITION
lrad	.false.	Long and short wave atmospheric radiation scheme based on Harshvardhan (*coamm.F, amodel.F*): **true** - radiative adjustments to θ computed when integration loop coincides with a **dtrad** time interval **false** - do not do above
lralee	.false.	Raleigh damping layer (*coamm.F*): **true** - damp (u,v,w,θ) to mean state in the top **nrdamp** levels **false** - do not do above
lsfcen	.false.	Surface energy budget (*sfcenr.F*): **true** - temporal changes in ground temperature and wetness based upon surface fluxes and radiation **false** - no temporal changes in ground temperature or wetness
lspong	.false.	Sponge layer (*coamm.F*): **true** - damp (u,v,w,θ) to smoothed values in the top **nrdamp** levels **false** - do not do above
ltanl	.false.	Temperature analysis (*anlfld.F, oi_anl.F*): **true** - perform analysis of initial temperature field **false** - do not do above
ltanls	.false.	Surface temperature analysis (*anlfld.F, oi_anl.F*): **true** - perform analysis of initial surface temperature field **false** - do not do above
ltke	.true.	Subgrid scale mixing (*afore.F*): **true** - temporal variations in TKE computed **false** - no temporal variations in TKE
ltopoa	.true.	Topography (*coama.F, insfc.F*): **true** - terrain field given by user or by data base **false** -terrain field set to zero
lvarad	.false.	Hydrostatic variational adjustment (*stdp2z.F, incrup.F*): **true** - adjust pressure and potential temperature perturbations **false** - do not do above
lvgeo	.false.	Geostrophic winds (*arhsu.F, arhsv.F*): **true** - include large scale pressure gradient force computed from **ugeo** and **vgeo** arrays **false** - do not do above

VAR	DEFAULT VALUE	DEFINITION
lvpert	.false.	Perturbation added to v-component (*ainit.F*): **true** - add perturbations **false** - do not do above
lwritu	.false.	Formatting of model input/output data: **true** - unformatted (32-bit IEEE) **false** - formatted (5e13.6)

Table D-1: Namelist input options for the TKE mixing length parameter **iamxgl**. Column one contains the variables and arrays in the TKE subroutine *afore.F* that are controlled by specifying values (1)-(5) for **iamxgl**. The **al0** is a user defined input and **al00** depends on the zero and first moments of TKE.

iamxgl=	(1)	(2)	(3)	(4)	(5)
al2= (diagnostically computed mixing length)	$\frac{.76\sqrt{e}}{\sqrt{\partial\theta/\partial z}}$ - (stable) **al1** - (unstable)	same **al2** as (1)	same **al2** as (1)	fn_a(blht, stab, TKE)	$\frac{al00 \bullet z\kappa}{(al00 + z\kappa)}$
al1= (used in **al2**; constant wrt time)	$\frac{al0 \bullet z\kappa}{(al0 + z\kappa)}$	$(\Delta x \Delta y \Delta z)^{1/3}$	**al0**•Δz	**al1** not used	**al1** not used
ald= (used in the TKE dissipation term)	**al2**	same **ald** as (1)	same **ald** as (1)	fn_b(blht, stab, TKE)	**al2**
coef= (used in the TKE dissipation term)	$0.19+0.74\frac{al2}{al1}$	same coef as (1)	same coef as (1)	0.125	0.170
reference	Mellor & Yamada 1974	Klemp & Wilhelmson 1978	---	Therry & Lacarrere 1983	Mellor & Yamada 1982

Appendix E: Three sample map projections, input parameters, and model domains

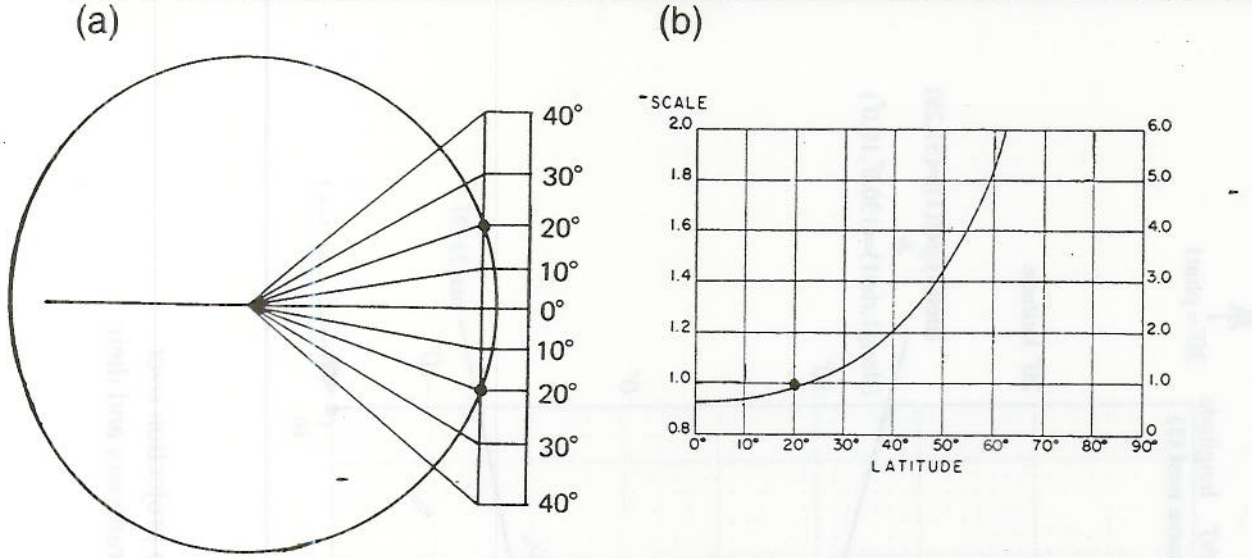


Figure E-1: Schematic representation of (a) a mercator map projection with standard latitude at 20°N (Haltiner and Williams, 1980), and (b) the corresponding graph showing variations in map scale factor with latitude (Saucier, 1955).

Table E-1: Namelist input parameters and values for the mercator map projection shown in Figure E-2.

Namelist Parameters	Definition and Use
alnnt = -----,	not used
delx = 120000.0,	coarse mesh (1) grid spacing in the x-direction
dely = 120000.0,	coarse mesh (1) grid spacing in the y-direction
iref(1) = 31,	coarse mesh (1) x-grid point corresponding to the reference latitude
ii(2) = 15,	coarse mesh (1) x-grid point corresponding to left boundary of inner nest (2)
ii(3) = ---,	not used
jref(1) = 28,	coarse mesh (1) y-grid point corresponding to the reference longitude
jj(2) = 15,	coarse mesh (1) y-grid point corresponding to left boundary of inner nest (2)
jj(3) = ---,	not used
ma(1) = 61,	coarse mesh (1) x-grid points
ma(2) = 91,	inner nest (2) x-grid points
ma(3) = ---,	inner nest (3) x-grid points
na(1) = 61,	coarse mesh (1) y-grid points
na(2) = 55,	inner nest (2) y-grid points
na(3) = ---,	inner nest (3) y-grid points
nnest = 2,	model domains
nproj = 1,	mecator map projection
phnt1 = 30,	latitude where map projection intersects the earth (i.e. map factor=1)
phnt2 = -----,	not used
rlat1 = 10.0,	reference latitude
rlong1 = 130.0,	reference longitude

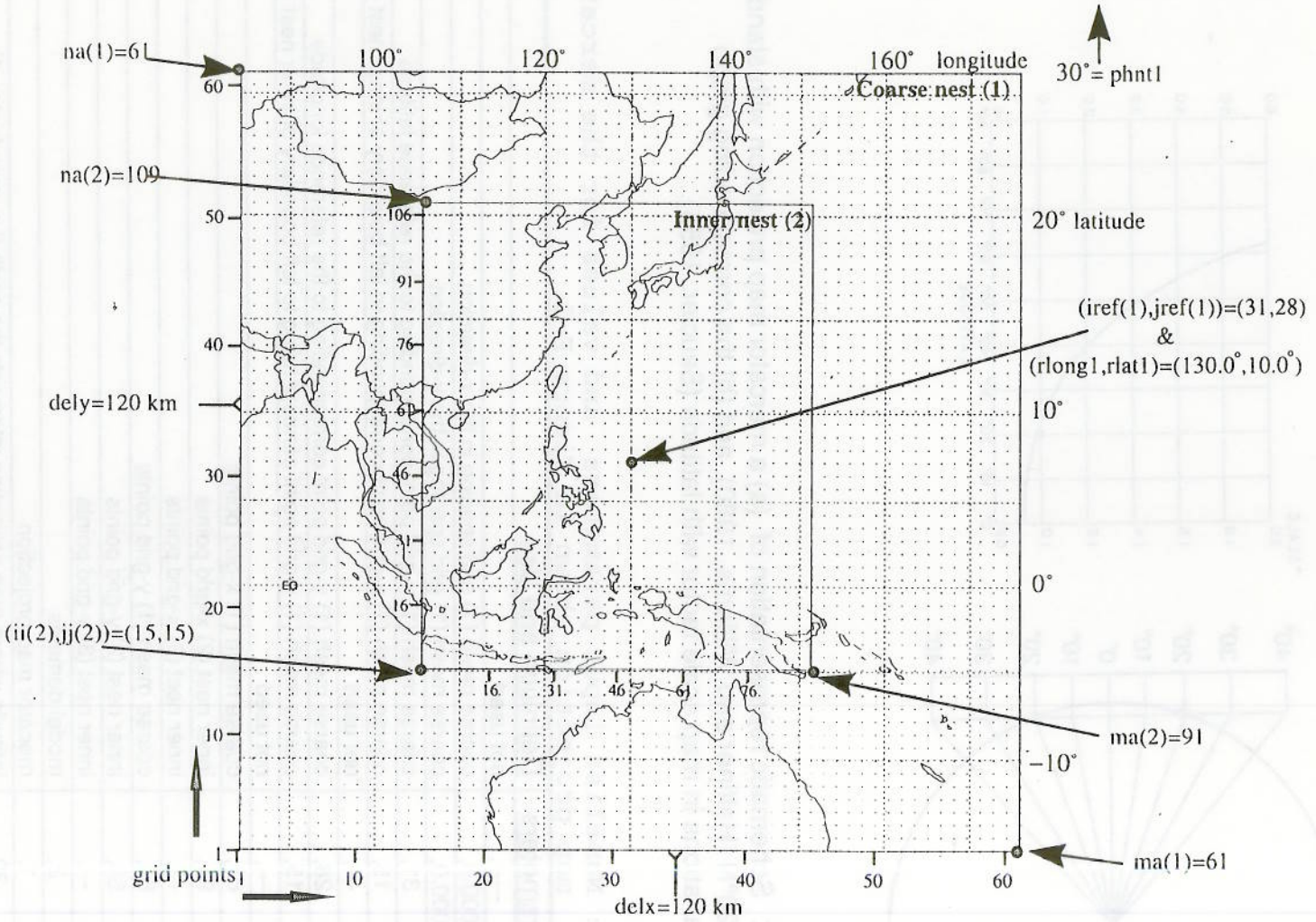


Figure E-2: Example of a COAMPS horizontal domain using a mercator map projection over the western Pacific. The arrows correlate locations on the grid to the input parameters and their user-specified value. These parameters are further defined in Table E-1.

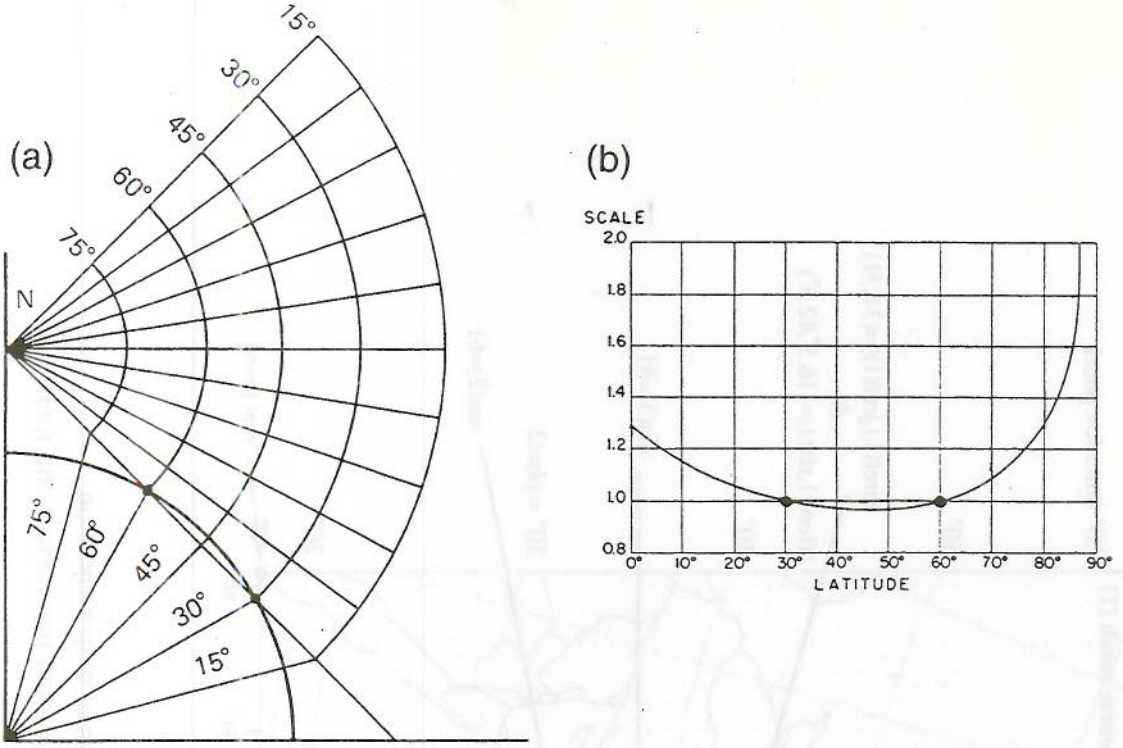


Figure E-3: Schematic representation of (a) a lambert conformal map projection with standard latitudes at 30°N and 60°N (Haltiner and Williams, 1980), and (b) the corresponding graph showing variations in map scale factor with latitude (Saucier, 1955).

Table E-2: Namelist input parameters and values for the lambert conformal map projection shown in Figure E-4.

Namelist Parameters	Definition and Use
alnnt = 16.5,	north/south longitude used to orient and align the grid
delx = 81000.0,	coarse mesh (1) grid spacing in the x-direction
dely = 81000.0,	coarse mesh (1) grid spacing in the y-direction
iref(1) = 31,	coarse mesh (1) x-grid point corresponding to the reference latitude
ii(2) = 22,	coarse mesh (1) x-grid point corresponding to left boundary of inner nest (2)
ii(3) = 16,	inner nest (2) x-grid point corresponding to left boundary of inner nest (3)
jref(1) = 31,	coarse mesh (1) y-grid point corresponding to the reference longitude
jj(2) = 22,	coarse mesh (1) y-grid point corresponding to left boundary of inner nest (2)
jj(3) = 16,	inner nest (2) y-grid point corresponding to left boundary of inner nest (3)
ma(1) = 61,	coarse mesh (1) x-grid points
ma(2) = 61,	inner nest (2) x-grid points
ma(3) = 91,	inner nest (3) x-grid points
na(1) = 61,	coarse mesh (1) y-grid points
na(2) = 61,	inner nest (2) y-grid points
na(3) = 91,	inner nest (3) y-grid points
nnest = 3,	model domains
nproj = 2,	lambert conformal map projection
phnt1 = 60.0,	latitude where map projection intersects the earth (i.e. map factor=1)
phnt2 = 30.0,	second latitude where map projection intersects the earth
rlat1 = 42.5,	reference latitude
rlong1 = 16.5,	reference longitude

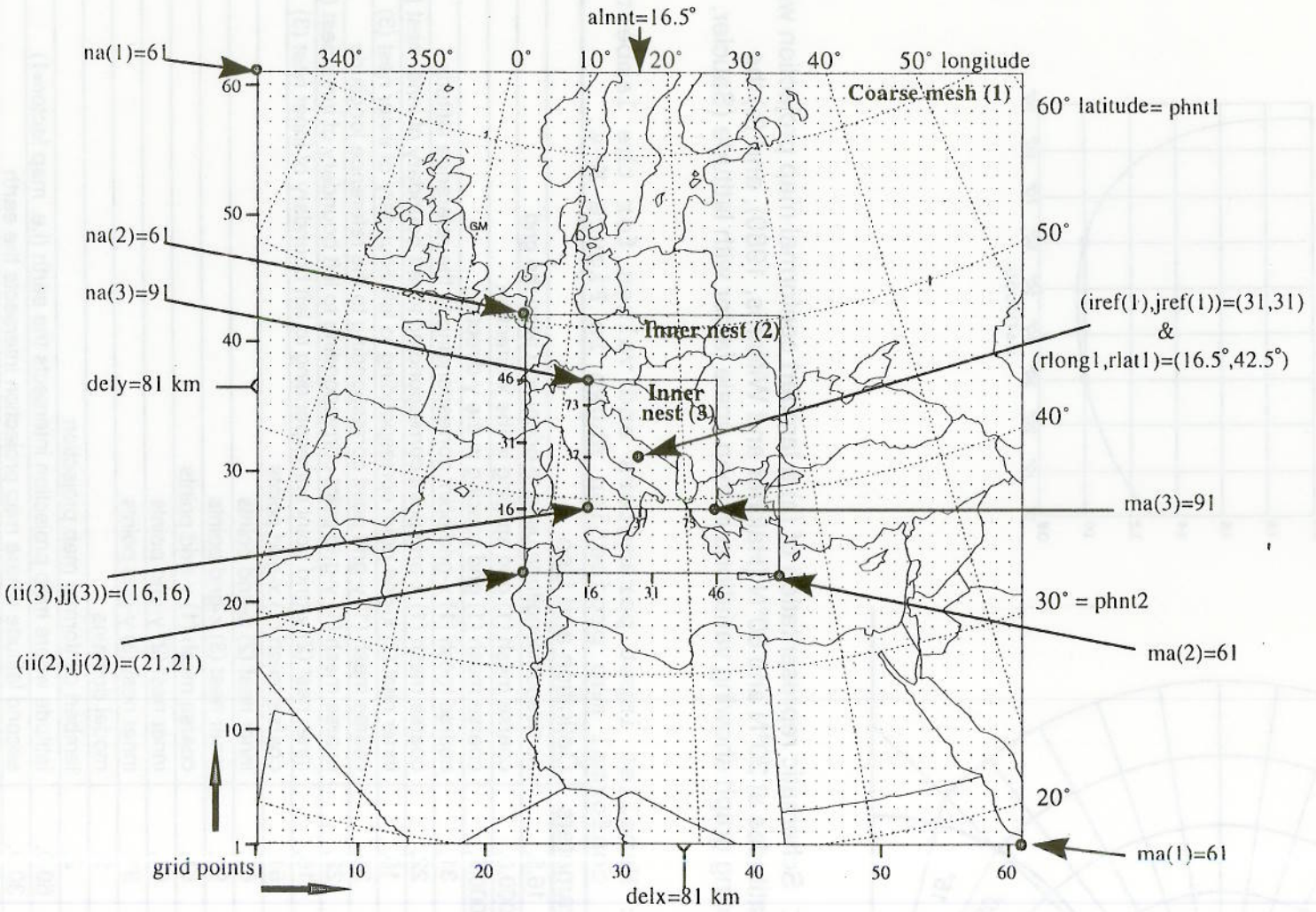


Figure E-4: Example of a COAMPS horizontal domain using a lambert conformal map projection centered over the Adriatic Sea. The arrows correlate locations on the grid to the input parameters and their user-specified value. These parameters are further defined in Table E-2.

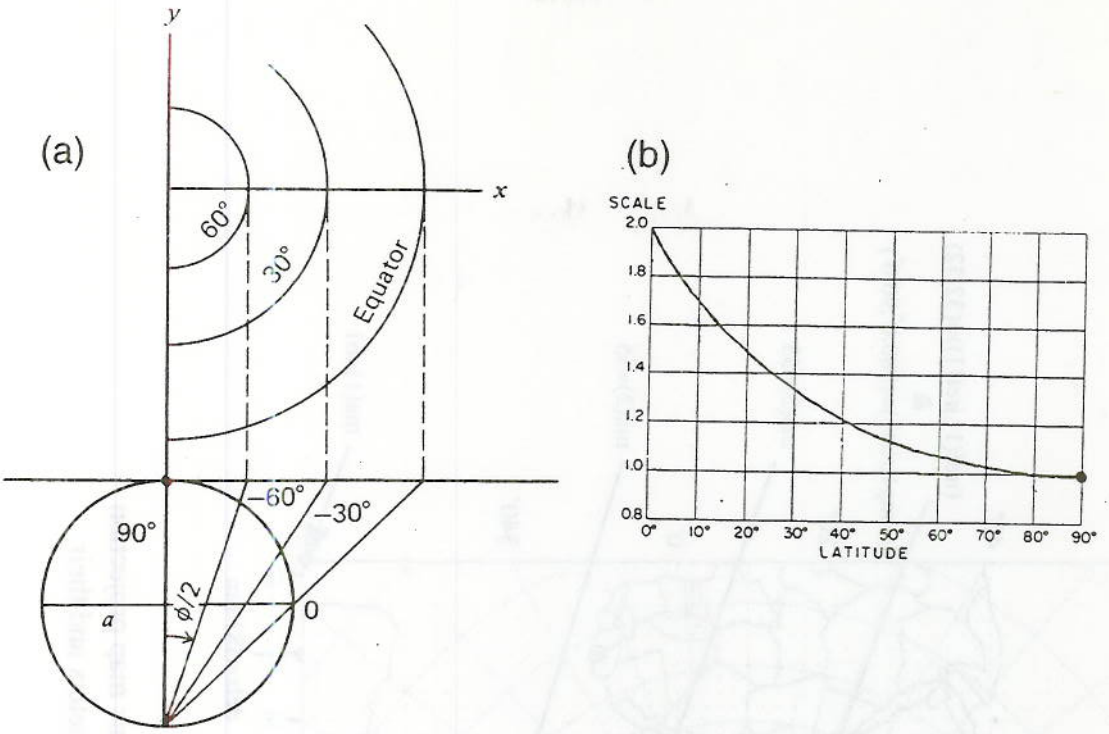


Figure E-5: Schematic representation of (a) a polar stereographic map projection with standard latitude at 90°N (Haltiner and Williams, 1980), and (b) the corresponding graph showing variations in map scale factor with latitude (Saucier, 1955).

Table E-3: Namelist input parameters and values for the polar stereographic map projection shown in Figure E-6.

Namelist Parameters			Definition and Use
alnnt	=	280.0,	north/south longitude used to orient and align the grid
delx	=	381000.0,	coarse mesh (1) grid spacing in the x-direction
dely	=	381000.0,	coarse mesh (1) grid spacing in the y-direction
ih1	=	32,	coarse mesh (1) x-grid point corresponding to the standard latitude
ii21	=	11,	coarse mesh (1) x-grid point corresponding to left boundary of inner nest (2)
ii32	=	21,	inner nest (2) x-grid point corresponding to left boundary of inner nest (3)
jh1	=	32,	coarse mesh (1) y-grid point corresponding to the standard longitude
jj21	=	11,	coarse mesh (1) y-grid point corresponding to left boundary of inner nest (2)
jj32	=	21,	inner nest (2) y-grid point corresponding to left boundary of inner nest (3)
ma1	=	61,	coarse mesh (1) x-grid points
ma2	=	55,	inner nest (2) x-grid points
ma3	=	55,	inner nest (3) x-grid points
na1	=	61,	coarse mesh (1) y-grid points
na2	=	55,	inner nest (2) y-grid points
na3	=	55,	inner nest (3) y-grid points
nnest	=	3,	model domains
nproj	=	3,	polar stereographic map projection
phnt1	=	60.0,	latitude where map projection intersects the earth (i.e. map factor=1)
phnt2	=	-----,	not used
rlat1	=	90.0,	standard latitude
rlong1	=	280.0,	standard longitude

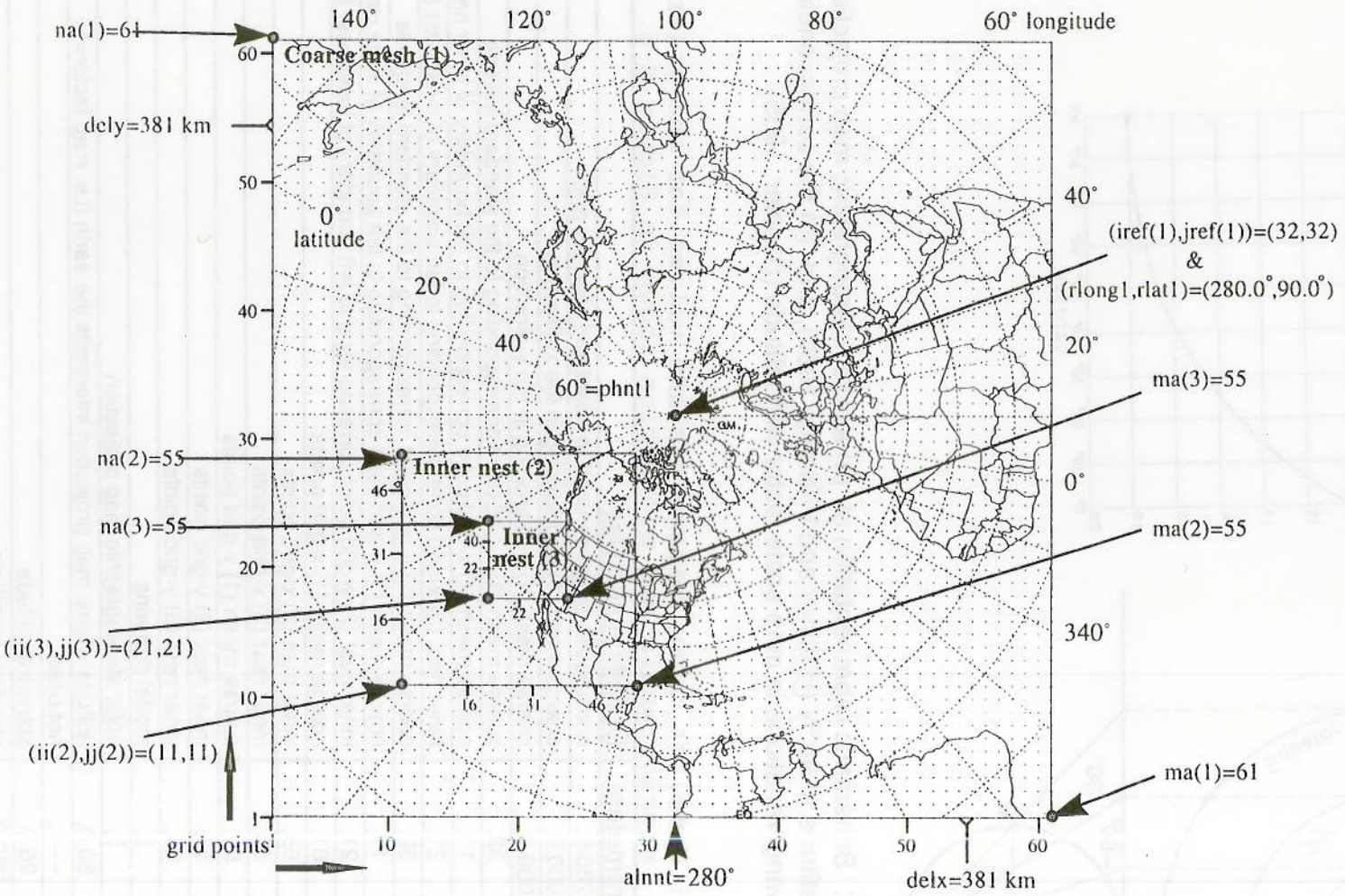


Figure E-6: Example of a COAMPS horizontal domain using a polar stereographic map projection over the North Pole. The arrows correlate locations on the grid to the input parameters and their user-specified value. These parameters are further defined in Table E-3.